

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4676140号
(P4676140)

(45) 発行日 平成23年4月27日(2011.4.27)

(24) 登録日 平成23年2月4日(2011.2.4)

(51) Int. Cl.	F I		
G 1 0 L 19/00 (2006.01)	G 1 0 L 19/00	2 1 3	
G 1 0 L 19/02 (2006.01)	G 1 0 L 19/02	1 4 2 F	
	G 1 0 L 19/02	1 5 0	

請求項の数 18 外国語出願 (全 64 頁)

(21) 出願番号	特願2003-309277 (P2003-309277)	(73) 特許権者	500046438
(22) 出願日	平成15年9月1日(2003.9.1)		マイクロソフト コーポレーション
(65) 公開番号	特開2004-264811 (P2004-264811A)		アメリカ合衆国 ワシントン州 9805
(43) 公開日	平成16年9月24日(2004.9.24)		2-6399 レッドモンド ワン マイ
審査請求日	平成18年8月23日(2006.8.23)		クロソフト ウエイ
(31) 優先権主張番号	60/408,517	(74) 代理人	100077481
(32) 優先日	平成14年9月4日(2002.9.4)		弁理士 谷 義一
(33) 優先権主張国	米国 (US)	(74) 代理人	100088915
(31) 優先権主張番号	10/642551		弁理士 阿部 和夫
(32) 優先日	平成15年8月15日(2003.8.15)	(72) 発明者	ナビーン サンプディ
(33) 優先権主張国	米国 (US)		アメリカ合衆国 98074 ワシントン
			州 サマリッシュ ノースイースト 23
			4 プレイス 518

最終頁に続く

(54) 【発明の名称】 オーディオの量子化および逆量子化

(57) 【特許請求の範囲】

【請求項1】

オーディオエンコーダにおいて、コンピュータにより実施される方法であって、
複数のチャンネルでオーディオデータを受け取ることと、
前記複数のチャンネルの各チャンネルを可変サイズウィンドウに分割することであって、前記複数のチャンネルの各チャンネルのウィンドウ構成は前記複数のチャンネルの他のチャンネルから独立していることと、

前記ウィンドウを複数のタイルにグループ化することであって、グループ化されたウィンドウは、各々のタイルごとに同一の開始位置および同一の終了位置を有することと、

前記複数のタイルの1つのタイルに対して、前記複数のチャンネルの各チャンネルに関するチャンネル固有量子化係数を適用すること、および前記1つのタイルに関するタイル量子化係数を適用することを含む、前記オーディオデータを量子化することと

を含むことを特徴とする方法。

【請求項2】

前記チャンネル固有量子化係数は、チャンネル固有量子化ステップ変更子であることを特徴とする請求項1に記載の方法。

【請求項3】

前記エンコーダは、前記複数のチャンネルにまたがって再構成品質のバランスをとるために複数のチャンネル固有量子化ステップ変更子を適用することを特徴とする請求項2に記載の方法。

【請求項 4】

前記エンコーダ内で、1つまたは複数の判断基準に少なくとも部分的に基づいて前記量子化係数を計算することをさらに含むことを特徴とする請求項 1 に記載の方法。

【請求項 5】

前記判断基準は、前記複数のチャンネルにまたがる再構成品質の均等を含むことを特徴とする請求項 4 に記載の方法。

【請求項 6】

前記判断基準は、前記複数のチャンネルのうちで知覚的に他のチャンネルより重要な1つまたは複数のチャンネルを優先することを含むことを特徴とする請求項 4 に記載の方法。

【請求項 7】

前記計算することは、前記複数のチャンネルのめいめいのエネルギーに少なくとも部分的に基づくことを特徴とする請求項 4 に記載の方法。

【請求項 8】

前記エンコーダ内で、オープンループ評価によって前記量子化係数を計算することをさらに含むことを特徴とする請求項 1 に記載の方法。

【請求項 9】

前記エンコーダ内で、クローズドループ評価によって前記量子化係数を計算することをさらに含むことを特徴とする請求項 1 に記載の方法。

【請求項 10】

オーディオデコーダにおいて、コンピュータにより実施される方法であって、
 複数のチャンネルでエンコードされたオーディオデータを受け取ることと、
1つまたは複数のタイルに対して、1つまたは複数のタイル量子化ステップサイズ、および複数のチャンネル固有クオンタイザステップ変更子の情報を検索することであって、前記1つまたは複数のタイルの各々は、前記複数のチャンネル中の異なるチャンネルに存在し、同一の開始位置および同一の終了位置を有する複数のウィンドウをグループ化すること

、
前記複数のタイルの1つのタイルに対して、逆量子化で前記1つのタイルに関する前記1つまたは複数のタイル量子化ステップサイズの1つを適用すること、および前記複数のチャンネルの各チャンネルに関する前記複数のチャンネル固有クオンタイザステップ変更子の1つを適用することを含む、前記オーディオデータをデコードすること
 を含むことを特徴とする方法。

【請求項 11】

前記複数のチャンネルは、2つのチャンネルからなることを特徴とする請求項 1 または 10 に記載の方法。

【請求項 12】

前記複数のチャンネルは、2つを超えるチャンネルからなることを特徴とする請求項 1 または 10 に記載の方法。

【請求項 13】

前記検索することは、前記複数のチャンネル固有クオンタイザステップ変更子の精度を示す複数のビットを入手することを含むことを特徴とする請求項 10 に記載の方法。

【請求項 14】

前記検索することは、変更子が0の値を有するかどうかを示すために変更子ごとに単一のビットを入手することを含むことを特徴とする請求項 10 に記載の方法。

【請求項 15】

前記タイル量子化係数は、タイル量子化ステップサイズであることを特徴とする請求項 1 に記載の方法。

【請求項 16】

前記検索することは、前記1つまたは複数のタイルの1つのタイルに対して、前記1つのタイルに関する初期タイル量子化ステップサイズの修正を示す複数のビットを入手することを含むことを特徴とする請求項 10 に記載の方法。

10

20

30

40

50

【請求項 17】

オーディオエンコーダにおいて、コンピュータに

複数のチャンネルでオーディオデータを受け取ることと、

前記複数のチャンネルの各チャンネルを可変サイズウィンドウに分割することであって、前記複数のチャンネルの各チャンネルのウィンドウ構成は前記複数のチャンネルの他のチャンネルから独立していることと、

前記ウィンドウを複数のタイルにグループ化することであって、グループ化されたウィンドウは、各々のタイルごとに同一の開始位置および同一の終了位置を有することと、

前記複数のタイルの1つのタイルに対して、前記複数のチャンネルの各チャンネルに関するチャンネル固有量子化係数を適用すること、および前記1つのタイルに関するタイル量子化係数を適用することを含む、前記オーディオデータを量子化することと

を実行させるためのプログラムを記録したことを特徴とするコンピュータ可読記録媒体

。

【請求項 18】

オーディオデコーダにおいて、コンピュータに

複数のチャンネルでエンコードされたオーディオデータを受け取ることと、

1つまたは複数のタイルに対して、1つまたは複数のタイル量子化ステップサイズ、および複数のチャンネル固有クオンタイザステップ変更子の情報を検索することであって、前記1つまたは複数のタイルの各々は、前記複数のチャンネル中の異なるチャンネルに存在し、同一の開始位置および同一の終了位置を有する複数のウィンドウをグループ化することと

前記複数のタイルの1つのタイルに対して、逆量子化で前記1つのタイルに関する前記1つまたは複数のタイル量子化ステップサイズの1つを適用すること、および前記複数のチャンネルの各チャンネルに関する前記複数のチャンネル固有クオンタイザステップ変更子の1つを適用することを含む、前記オーディオデータをデコードすることと

を実行させるためのプログラムを記録したことを特徴とするコンピュータ可読記録媒体

。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、エンコーディングおよびデコーディングでのオーディオ情報の処理に関する。具体的には、本発明は、オーディオエンコーディングおよびオーディオデコーディングでの量子化および逆量子化に関する。

【背景技術】

【0002】

コンパクトディスク、デジタル無線電話網、およびインターネットを介するオーディオ配信の導入に伴って、デジタルオーディオがありふれたものになってきた。技術者は、さまざまな手法を使用して、デジタルオーディオの品質を維持しながら、デジタルオーディオを効率的に処理する。これらの手法を理解するために、コンピュータでオーディオ情報がどのように表現され、処理されるかを理解することが役に立つ。

【0003】

I. コンピュータでのオーディオ情報の表現

コンピュータでは、オーディオ情報を、オーディオ情報を表す一連の数として処理する。たとえば、単一の数が、オーディオサンプルを表すことができ、このオーディオサンプルは、特定の時刻の振幅値（すなわち、音の大きさ）である。サンプル深さ（sample depth）、サンプリングレート（sampling rate）、およびチャンネルモードなど、複数の要因が、オーディオ情報の品質に影響する。

【0004】

サンプル深さ（または精度）は、サンプルを表すのに使用される数の範囲を示す。サンプルに可能な値が多くなれば、振幅のより微妙な変動を取り込めるようになるので、品質

10

20

30

40

50

が高くなる。たとえば、8ビットサンプルは、256個の可能な値を有するが、16ビットサンプルは、65536個の可能な値を有する。24ビットサンプルでは、普通の音の大きさの変動を非常に微細に取り込むことができ、異常に大きい音も取り込むことができる。

【0005】

サンプリングレート（通常は、サンプル数毎秒として測定される）も、品質に影響する。サンプリングレートが高いほど、より高い周波数の音を表現できるので、品質が高くなる。一般的なサンプリングレートは、8000、11025、22050、32000、44100、48000、および96000サンプル毎秒である。

【0006】

モノラルおよびステレオが、オーディオの2つの一般的なチャンネルモードである。モノラルモードでは、オーディオ情報が、1つのチャンネルに存在する。ステレオモードでは、オーディオ情報が、通常は左チャンネルおよび右チャンネルと称する2つのチャンネルに存在する。5.1チャンネル、7.1チャンネル、または9.1チャンネルのサラウンドサウンド（「1」は、サブウーファ（副低音スピーカ）または低周波数効果チャンネルを示す）などの、より多くのチャンネルを有する他のモードも可能である。表1に、対応する生ビットレートコストと共に、異なる品質レベルのオーディオの複数のフォーマットを示す。

【0007】

【表1】

品質	サンプル深さ (ビット/サンプル)	サンプリングレート (サンプル/秒)	モード	生ビットレート (ビット/秒)
インター ネット電話	8	8,000	モノラル	64,000
電話	8	11,025	モノラル	88,200
CD オーディオ	16	44,100	ステレオ	1,411,200

表1：異なる品質のオーディオ情報のビットレート

【0008】

サラウンドサウンドオーディオは、通常は、さらに高い生ビットレートを有する。表1からわかるように、高品質オーディオ情報のコストは、高いビットレートである。高品質オーディオ情報は、大量のコンピュータストレージ（記憶装置）および伝送容量を消費する。しかし、企業および消費者は、高品質マルチチャンネルオーディオコンテンツを作成し、配布し、再生するのに、ますますコンピュータに頼る。

【0009】

II. コンピュータでのオーディオ情報の処理

多くのコンピュータおよびコンピュータネットワークに、生デジタルオーディオを処理するリソース（資源）が不足している。圧縮（エンコーディングまたはコーディングとも称する）では、情報をよりビットレートの低い形に変換することによって、オーディオ情報の保管および伝送のコストを下げる。圧縮は、ロスレス（損失のない）（lossless）（品質に影響しない）、またはロッシイ（損失のある）（lossy）（品質に影響するが、後続のロスレス圧縮からのビットレート削減がより劇的である）とすることができる。圧縮解除（復元）（デコーディング（複合化、解読、逆符号化）とも称する）によって、圧縮された形から元の情報の再構成されたバージョンが抽出される。

【0010】

A. 標準的な知覚オーディオエンコーダおよびデコーダ

一般に、オーディオ圧縮の目標は、オーディオ信号をデジタルに表現して、可能な最小限の量のビットで最高の信号品質をもたらすことである。通常のオーディオエンコーダ/デコーダ[「コーデック(codex)」]システムでは、サブバンド/変換コーディング、量子化、レート制御、および可変長コーディングを使用して、その圧縮を達成する

10

20

30

40

50

。量子化および他のロッシイ圧縮手法によって、潜在的に可聴の雑音がオーディオ信号に導入される。雑音の可聴性は、どれほどの雑音があるかと、雑音のどれだけを聴取者が知覚するか依存する。第1の要因は、主に客観的な品質に関し、第2の要因は、人間による音の知覚に依存する。

【0011】

図1に、従来技術による変換ベースの知覚オーディオエンコーダ(100)の一般化された図を示す。図2に、従来技術による対応するオーディオデコーダ(200)の一般化された図を示す。図1および2に示されたコーデックシステムは、一般化されているが、Microsoft Corporation社のWindows(登録商標)Media Audio[「WMA」]エンコーダおよびデコーダの諸バージョンを含む、複数の実世界のコーデックシステムに見られる特性を有する。他のコーデックシステムは、Motion Picture Experts Group、Audio Layer 3[「MP3」]標準規格、Motion Picture Experts Group 2、Advanced Audio Coding[「AAC」]標準規格、およびDolby AC3によって提供されるか指定される。コーデックシステムに関する追加情報については、めいめいの標準規格または技術的刊行物を参照されたい。

【0012】

1. 知覚オーディオエンコーダ

全体として、エンコーダ(100)は、入力オーディオサンプル(105)の時系列を受け取り、オーディオサンプル(105)を圧縮し、エンコーダ(100)のさまざまなモジュールによって作られる情報を多重化して、ビットストリーム(195)を出力する。エンコーダ(100)には、周波数トランスフォーマ(変換器)(frequency transformer)(110)、マルチチャネルトランスフォーマ(multi-channel transformer)(120)、知覚モデラ(モデル信号発生器)(perception modeler)(130)、ウェイト(重み付け器)(weighter)(140)、クオンタイザ(量子化器)(quantizer)(150)、エントロピーエンコーダ(entropy encoder)(160)、コントローラ(170)、およびビットストリームマルチプレクサ(bitstream multiplexer)[「MUX」](180)が含まれる。

【0013】

周波数トランスフォーマ(110)は、オーディオサンプル(105)を受け取り、周波数領域のデータに変換する。たとえば、周波数トランスフォーマ(110)は、オーディオサンプル(105)をブロックに分割し、このブロックは、可変時間分解能を可能にするために可変サイズを有することができる。小さいブロックを用いると、入力オーディオサンプル(105)の短いアクティブな推移セグメント(区間)で時間詳細をより多く保存できるようになるが、ある程度周波数分解能が犠牲になる。対照的に、大きいブロックは、よりよい周波数分解能とより悪い時間分解能を有し、通常は、より長い少数のアクティブセグメント(区間)でのより高い圧縮効率が可能になる。ブロックをオーバーラップさせて、そうでなければ後の量子化によって導入される可能性があるブロックの間の知覚的不連続性を減らすことができる。マルチチャネルオーディオについて、周波数トランスフォーマ(110)では、特定のフレーム内のチャネルごとに同一のパターンのウィンドウが使用される。周波数トランスフォーマ(110)は、周波数係数データのブロックをマルチチャネルトランスフォーマ(120)に出力し、ブロックサイズなどのサイド情報をMUX(180)に出力する。

【0014】

マルチチャネルオーディオデータの場合に、周波数トランスフォーマ(110)によって作られる周波数係数データの複数のチャンネルが、しばしば関連する。この関連を活用するために、マルチチャネルトランスフォーマ(120)によって、複数のオリジナルの独立にコーディングされたチャンネルを、連繋して(一緒に)コーディングされたチャンネルに変換することができる。たとえば、入力がステレオモードである場合に、マルチチャネル

トランスフォーマ(120)によって、左右のチャンネルを和と差のチャンネルに変換することができる。

【0015】

【数1】

$$X_{\text{Sum}}[k] = \frac{X_{\text{Left}}[k] + X_{\text{Right}}[k]}{2} \quad (1)$$

$$X_{\text{Diff}}[k] = \frac{X_{\text{Left}}[k] - X_{\text{Right}}[k]}{2} \quad (2)$$

【0016】

または、マルチチャンネルトランスフォーマ(120)によって、左右のチャンネルを独立にコーディングされたチャンネルとして渡すことができる。独立にコーディングされたチャンネルまたは連繋してコーディングされたチャンネルの使用の判断は、事前に決定するか、エンコーディング中に適応的に行うことができる。たとえば、エンコーダ(100)によって、(a)マルチチャンネル変換ありおよびなしのコーディングチャンネルの間のエネルギー分離と、(b)左右の入力チャンネルの間の励起パターンの不一致を考慮するオープンループ選択判断を用いて、ステレオチャンネルを連繋してまたは独立にのどちらでコーディングするかを決定する。そのような判断は、ウィンドウごとの基準で行うか、判断を単純にするためにフレームごとに1回だけ行うことができる。マルチチャンネルトランスフォーマ(120)は、使用されるチャンネルモードを示すサイド情報をMUX(180)に出力する。

【0017】

エンコーダ(100)は、マルチチャンネル変換の後に、オーディオデータのブロックにマルチチャンネル再行列化を適用することができる。連繋してコーディングされたチャンネルの低ビットレートのマルチチャンネルオーディオデータについて、エンコーダ(100)は、あるチャンネル(たとえば差チャンネル)の情報を選択的に抑圧して、残りのチャンネル(たとえば和チャンネル)の品質を改善する。たとえば、エンコーダ(100)は、スケーリング係数によって差チャンネルをスケーリングする。

【0018】

【数2】

$$\tilde{X}_{\text{Diff}}[k] = \rho \cdot X_{\text{Diff}}[k] \quad (3)$$

【0019】

ここで、 ρ の値は、(a)雑音興奮比率(Noise to Excitation Ratio) [「NER」]などの知覚オーディオ品質測定値の現在の平均レベルと、(b)仮想バッファの現在の満杯度と、(c)エンコーダ(100)のビットレートおよびサンプリングレート設定と、(d)左右の入力チャンネルのチャンネルセパレーションとに基づく。

【0020】

知覚モデル(130)は、人間の聴覚系のモデルに従ってオーディオデータを処理して、所与のビットレートの再構成されたオーディオ信号の知覚される品質を改善する。たとえば、聴覚モデルでは、通常、人間の聴取帯および臨界帯域の範囲が考慮される。人間の神経系では、周波数のサブレンジが統合される。この理由から、聴覚モデルでは、臨界帯域によってオーディオ情報を編成し、処理することができる。異なる聴覚モデルでは、異なる数の臨界帯域(たとえば、25個、32個、55個、または109個)および/または臨界帯域の異なるカットオフ周波数が使用される。バークバンド(bark band: 叫び声の帯域)が、臨界帯域の周知の例である。範囲および臨界帯域のほかに、オーディオ信号の間の相互作用が、知覚に劇的に影響する可能性がある。単独で提示される場合に明瞭に聴取可能であるオーディオ信号が、マスクまたはマスクング信号と称する別のオーディオ信号が存在すると完全に聴取不能になる可能性がある。人間の耳は、マスクングされる信号のひずみまたは他の忠実度の消失(すなわち雑音)に比較的鈍感であり、した

10

20

30

40

50

がって、マスキングされる信号に、知覚されるオーディオ品質を劣化させずにより多くのひずみを含めることができる。さらに、聴覚モデルでは、人間による音の知覚の物理的態様または神経的態様に関するさまざまな他の要因を考慮することができる。

【0021】

知覚モデル(130)は、雑音の可聴性を減らすためにオーディオデータの雑音を整形するのにウェイタ(140)が使用する情報を入力する。たとえば、さまざまな手法のいずれかを使用して、ウェイタ(140)は、受け取った情報に基づいて量子化行列(時々、マスクと称する)の重みづけ係数(時々、スケーリング係数と称する)を生成する。量子化行列の重みづけ係数には、オーディオデータ内の複数の量子化帯域ごとの重みが含まれ、量子化帯域は、周波数係数の周波数範囲である。量子化帯域の数は、臨界帯域の数以下とすることができる。したがって、重みづけ係数によって、雑音が量子化帯域にまたがって分散する特性が示され、より多くの雑音をより聴取可能でない帯域に置き、より少ない雑音をより聴取可能な帯域に置くことによって、雑音の可聴性を最小にすることが目標になる。重みづけ係数は、振幅およびブロックからブロックへの量子化帯域の数で変化する可能性がある。ウェイタ(140)は、マルチチャネルトランスフォーマ(120)から受け取ったデータに重みづけ係数を適用する。

10

【0022】

一実施形態で、ウェイタ(140)は、マルチチャネルオーディオの各チャネルのウィンドウごとに重みづけ係数の組を生成するか、連繋してコーディングされたチャネルの並列ウィンドウについて重みづけ係数の単一の組を共用する。ウェイタ(140)は、係数データの重みづけされたブロックをクオンタイザ(150)に出力し、重みづけ係数の組などのサイド情報をMUX(180)に出力する。

20

【0023】

重みづけ係数の組を、直接圧縮を使用して、より効率的な表現のために圧縮することができる。直接圧縮手法では、エンコーダ(100)が、量子化行列の各要素を均一に量子化する。エンコーダは、量子化された要素を、行列の前の要素に対して相対的に差分コーディングし、差分コーディングされた要素をハフマンコーディングする。いくつかの場合に(たとえば、特定の量子化帯域の係数のすべてが、0の値に量子化されるか切り詰められる時)、デコーダ(200)は、すべての量子化帯域について重みづけ係数を必要としない。そのような場合には、エンコーダ(100)は、系列の次に必要な重みづけ係数の値と同一の値を、1つまたは複数の不要な重みづけ係数に与え、これによって、量子化行列の要素の差分コーディングをより効率的にする。

30

【0024】

あるいは、低ビットレートアプリケーションについて、エンコーダ(100)は、量子化行列をパラメータ圧縮して、たとえば量子化行列から計算される擬似自己相関パラメータの線形予測コーディング[「LPC」]を使用して、パラメータの組として量子化行列を表現することができる。

【0025】

クオンタイザ(150)は、ウェイタ(140)の出力を量子化し、エントロピエンコーダ(160)への量子化された係数データと、MUX(180)への量子化ステップサイズを含むサイド情報とを作る。量子化では、入力値の範囲を単一の値に写像し、情報の不可逆的な消失が導入されるが、量子化によって、エンコーダ(100)が、コントローラ(170)と共に、ビットストリーム(195)出力の品質およびビットレートを調整できるようになる。図1では、クオンタイザ(150)が、適応均一スカラクオンタイザである。クオンタイザ(150)は、各周波数係数に同一の量子化ステップサイズを適用するが、量子化ステップサイズ自体を、量子化ループのある反復から次の反復の間で変更して、エントロピエンコーダ(160)出力のビットレートに影響を及ぼすことができる。他の種類の量子化が、不均一ベクトル量子化および/または非適応量子化である。

40

【0026】

エントロピエンコーダ(160)は、クオンタイザ(150)から受け取る量子化され

50

た係数データをロスレス圧縮する。エントロピエンコーダ(160)は、オーディオ情報のエンコーディングに費やされるビット数を計算し、この情報をレート/品質コントローラ(170)に渡すことができる。

【0027】

コントローラ(170)は、クオンタイザ(150)と一緒に働いて、エンコーダ(100)の出力のビットレートおよび/または品質を調整する。コントローラ(170)は、エンコーダ(100)の他のモジュールから情報を受け取り、受け取った情報を処理して、現在の条件での所望の量子化ステップサイズを判定する。コントローラ(170)は、ビットレート制約および品質制約を満足することを目標に、量子化ステップサイズをクオンタイザ(150)に出力する。

10

【0028】

エンコーダ(100)は、オーディオデータのブロックに雑音置換および/または帯域切詰を適用することができる。低ビットレートおよび中ビットレートで、オーディオエンコーダ(100)は、雑音置換を使用して、ある帯域の情報を伝える。帯域切詰では、ブロックの測定された品質から低い品質が示される場合に、エンコーダ(100)が、ある(通常はより高い周波数の)帯域の係数を完全に除去して、残りの帯域の総合的な品質を改善することができる。

【0029】

MUX(180)は、オーディオエンコーダ(100)の他のモジュールから受け取るサイド情報を、エントロピエンコーダ(160)から受け取ったエントロピエンコーディングされたデータと多重化する。MUX(180)は、オーディオデコーダが認識するフォーマットで情報を出力する。MUX(180)には、オーディオの複雑さの変化に起因するビットレートの短期間変動を平滑化するために、エンコーダ(100)によって出力されるビットストリーム(195)を保管する仮想バッファが含まれる。

20

【0030】

2. 知覚オーディオデコーダ

全体として、デコーダ(200)は、エントロピエンコードされたデータならびにサイド情報を含む圧縮オーディオ情報のビットストリーム(205)を受け取り、このビットストリームから、オーディオサンプル(295)を再構成する。オーディオデコーダ(200)には、ビットストリームデマルチプレクサ[「DEMUX」](210)、エントロピデコーダ(220)、逆クオンタイザ(230)、雑音ジェネレータ(240)、逆ウェイト(250)、逆マルチチャネルトランスフォーマ(260)、および逆周波数トランスフォーマ(270)が含まれる。

30

【0031】

DEMUX(210)は、ビットストリーム(205)の情報を解析し、情報をデコーダ(200)のモジュールに送る。DEMUX(210)には、オーディオの複雑さの変動、ネットワークジッタ、および/または他の要因に起因するビットレートの短期間変動を補償するために、1つまたは複数のバッファが含まれる。

【0032】

エントロピデコーダ(220)は、DEMUX(210)から受け取ったエントロピコードをロスレス圧縮解除し、量子化された周波数係数データを作る。エントロピデコーダ(220)は、通常は、エンコーダで使用されるエントロピエンコーディング手法の逆を適用する。

40

【0033】

逆クオンタイザ(230)は、DEMUX(210)から量子化ステップサイズを受け取り、エントロピデコーダ(220)から量子化された周波数係数データを受け取る。逆クオンタイザ(230)は、量子化された周波数係数データに量子化ステップサイズを適用して、周波数係数データを部分的に再構成する。

【0034】

雑音ジェネレータ(240)は、DEMUX(210)から、データのブロックのどの

50

帯域が雑音置換されたかを示す情報と、雑音の形に関するパラメータを受け取る。雑音ジェネレータ(240)は、示された帯域のパターンを生成し、その情報を逆ウェイト(250)に渡す。

【0035】

逆ウェイト(250)は、DEMUX(210)から重みづけ係数を受け取り、雑音ジェネレータ(240)から雑音置換された帯域のパターンを受け取り、逆クオンタイザ(230)から部分的に再構成された周波数係数データを受け取る。必要に応じて、逆ウェイト(250)は、たとえば、量子化された行列の要素のエントロピデコーディング、逆差分コーディング、および逆量子化などによって、重みづけ係数を圧縮解除する。逆ウェイト(250)は、雑音置換されなかった帯域の部分的に再構成された周波数係数データに、重みづけ係数を適用する。その後、逆ウェイト(250)は、雑音置換された帯域に関する雑音ジェネレータ(240)から受け取った雑音パターンを加える。

10

【0036】

逆マルチチャンネルトランスフォーマ(260)は、逆ウェイト(250)から再構成された周波数係数データを受け取り、DEMUX(210)からチャンネルモード情報を受け取る。マルチチャンネルオーディオが、独立にコーディングされたチャンネルにある場合には、逆マルチチャンネルトランスフォーマ(260)は、チャンネルをそのまま通す。マルチチャンネルデータが、連繋してコーディングされたチャンネルにある場合には、逆マルチチャンネルトランスフォーマ(260)は、そのデータを独立にコーディングされたチャンネルに変換する。

20

【0037】

逆周波数トランスフォーマ(270)は、マルチチャンネルトランスフォーマ(260)によって出力された周波数係数データならびにDEMUX(210)からのブロックサイズなどのサイド情報を受け取る。逆周波数トランスフォーマ(270)は、エンコーダで使用される周波数変換の逆を適用し、再構成されたオーディオサンプル(295)のブロックを出力する。

【0038】

【非特許文献1】Yang et al., "An Inter-Channel Redundancy Removal Approach for High-Quality Multichannel Audio Compression," AES 109th Convention, Los Angeles, September 2000 ["Yang"]

30

【非特許文献2】Wang et al., "A Multichannel Audio Coding Algorithm for Inter-Channel Redundancy Removal," AES 110th Convention, Amsterdam, Netherlands, May 2001 ["Wang"]

【非特許文献3】Kuo et al., "A Study of Why Cross Channel Prediction Is Not Applicable to Perceptual Audio Coding," IEEE Signal Proc. Letters, vol. 8, no. 9, September 2001

【非特許文献4】Rao et al., Discrete Cosine Transform, Academic Press (1990)

【非特許文献5】Vaidyanathan, Multirate Systems and Filter Banks, Chapter 14.6, "Factorization of Unitary Matrices," Prentice Hall (1993)

【発明の開示】

40

【発明が解決しようとする課題】

【0039】

B. 標準的な知覚オーディオエンコーダおよび知覚オーディオデコーダの短所

上で説明した知覚エンコーダおよび知覚デコーダは、多くの応用例について良好な総合性能を有するが、複数の短所、特にマルチチャンネルオーディオの圧縮および圧縮解除に関する短所を有する。この短所によって、いくつかの場合、たとえば使用可能なビットレートが、入力オーディオチャンネルの数に対して少ない時に、再構成されるマルチチャンネルオーディオの品質が制限される。

【0040】

1. マルチチャンネルオーディオのフレーム区分での柔軟性のなさ

50

さまざまな点で、図1のエンコーダ(100)によって実行されるフレーム区分は、柔軟でない。

【0041】

前に述べたように、周波数トランスフォーマ(110)は、入力オーディオサンプル(105)のフレームを、周波数変換のために1つまたは複数のオーバーラップするウィンドウに分割するが、大きいウィンドウは、よりよい周波数分解能および冗長性除去をもたらす。小さいウィンドウは、よりよい時間分解能をもたらす。よりよい時間分解能は、信号が低エネルギーから高エネルギーに推移する時に導入される可聴プリエコー(pre-echo)アーチファクトを制御するのに役立つが、小さいウィンドウを使用すると、圧縮可能性が下がるので、エンコーダは、ウィンドウサイズを選択する時に、これらの考慮事項のバランスをとらなければならない。マルチチャンネルオーディオについて、周波数トランスフォーマ(110)は、フレームのチャンネルを同一の形で(すなわち、チャンネルでの同一のウィンドウ構成)区分するが、これは、図3aから3cに示されているように、いくつかの場合に非効率的である場合がある。

10

【0042】

図3aに、例のステレオオーディオ信号の波形(300)を示す。チャンネル0の信号には、推移アクティビティが含まれ、チャンネル1の信号は、相対的に静止している。エンコーダ(100)は、チャンネル0の信号推移を検出し、プリエコーを減らすために、フレームを、図3bに示された、より小さいオーバーラップする変調されたウィンドウ(301)に分割する。図を単純にするために、図3cでは、オーバーラップするウィンドウ構成(302)をボックス(箱)で示し、破線によってフレーム境界を示す。後の図も、この規約に従う。

20

【0043】

すべてのチャンネルに同一のウィンドウ構成をとらせることの短所は、1つまたは複数のチャンネルの静止信号(たとえば図3aから3cのチャンネル1)が、より小さいウィンドウに分割され、コーディング利得が下がる可能性があることである。その代わりに、エンコーダ(100)が、すべてのチャンネルにより長いウィンドウを使用させることができるが、推移を有する1つまたは複数のチャンネルにプリエコーが導入される。この問題は、複数のチャンネルをコーディングしなければならない時に悪化する。

30

【0044】

AAC(適応オーディオコーディング)を用いると、マルチチャンネル変換の対単位のチャンネルのグループ化が可能になる。左、右、中央、左後ろ、右後ろのチャンネルの中から、たとえば、左チャンネルと右チャンネルをステレオコーディングのためにグループ化し、左後ろチャンネルと右後ろチャンネルをステレオコーディングのためにグループ化することができる。異なるグループが、異なるウィンドウ構成を有することができるが、特定のグループの両方のチャンネルが、ステレオコーディングが使用される場合に同一のウィンドウ構成を有する。これによって、AACシステムでのマルチチャンネル変換の区分の柔軟性が制限され、対単位のみでのグループ化の使用についても同様である。

【0045】

2. マルチチャンネル変換での柔軟性のなさ

40

図1のエンコーダ(100)では、あるチャンネル間冗長性(inter-channel redundancy)が活用されるが、マルチチャンネル変換に関するさまざまな点で柔軟性がない。エンコーダ(100)を用いると、2種類の変換すなわち、(a)恒等変換(変換なしと同等である)、または(b)ステレオ対の和-差コーディングが可能である。これらの制限によって、3つ以上のチャンネルのマルチチャンネルコーディングが制約される。3つ以上のチャンネルを扱うことができるAACにおいても、マルチチャンネル変換は、1時に1対のチャンネルだけに制限される。

【0046】

複数のグループが、サラウンドサウンドチャンネルに関するマルチチャンネル変換に関して実験した(たとえば、非特許文献1(以下「Yang」)、非特許文献2(以下「Wan

50

g」)参照)。Yangのシステムでは、よい圧縮係数に関してチャンネルを相関解除 (decorrelate) するために、チャンネルにまたがるKarhunen - Loeve変換 [「KLT」] が使用される。Wangのシステムでは、整数対整数離散コサイン変換 (Discrete Cosine Transform) [「DCT」] が使用される。両方のシステムで、よい結果が与えられるが、まだ複数の制限がある。

【0047】

第1に、オーディオサンプルにKLTを使用する (Yangのシステムのように時間領域または周波数領域で) と、再構成で導入されるひずみが制御されない。YangのシステムのKLTは、マルチチャンネルオーディオの知覚オーディオコーディングに成功裡に使用されない。Yangのシステムでは、逆マルチチャンネル変換での、1つの (たとえば激しく量子化される) コーディングされたチャンネルから複数の再構成されるチャンネルへの漏れの量が制御されない。この短所は、文献で指摘されている (たとえば、非特許文献3参照)。言い換えると、あるコーディングされたチャンネルで「可聴でない」量子化が、複数の再構成されたチャンネルに分散される時に可聴になる可能性がある。というのは、逆重みづけが、逆マルチチャンネル変換の前に実行されるからである。Wangのシステムでは、マルチチャンネル変換を、エンコーダ内で重みづけおよび量子化の後に配置する (かつ、逆マルチチャンネル変換を、デコーダ内で逆量子化および逆重みづけの前に配置する) ことによって、この問題が克服される。しかし、Wangのシステムは、さまざまな他の短所を有する。マルチチャンネル変換の前に量子化を実行することは、マルチチャンネル変換を整数対整数にしなければならない、可能な変換の数が制限され、チャンネルにまたがる冗長性除去が制限されることを意味する。

【0048】

第2に、Yangのシステムは、KLT変換に制限される。KLT変換は、圧縮されるオーディオデータに適応されるが、Yangのシステムの、異なる種類の変換を使用する柔軟性は、制限されている。同様に、Wangのシステムでは、マルチチャンネル変換に整数対整数DCTが使用されるが、これは、エネルギーコンパクト化に関して通常のDCTほど良好ではなく、Wangのシステムの、異なる種類の変換を使用する柔軟性は、制限されている。

【0049】

第3に、YangのシステムおよびWangのシステムには、どのチャンネルを一緒に変換するかを制御する機構がなく、マルチチャンネル変換の異なる時に異なるチャンネルを選択的にグループ化する機構もない。そのような制御は、まったく互換性がないチャンネルにまたがるコンテンツの漏れを制限するのに役立つ。さらに、全体的に互換性のあるチャンネルであっても、ある期間にわたって互換性がなくなる場合がある。

【0050】

第4に、Yangのシステムでは、マルチチャンネル変換に、周波数帯域レベルでマルチチャンネル変換を適用するか否かに対する制御が欠けている。全体的に互換性のあるチャンネルの間であっても、それらのチャンネルが、ある周波数またはある周波数帯域で互換性がない場合がある。同様に、図1のエンコーダ (100) のマルチチャンネル変換には、サブチャンネルレベルでの制御が欠けており、どの帯域の周波数係数データをマルチチャンネル変換するかが制御されず、入力チャンネルの周波数帯域のうちに相関しないものがある時に生じる可能性がある非効率性が無視される。

【0051】

第5に、ソースチャンネルに互換性がある時であっても、しばしば、一緒に変換されるチャンネルの数を制御して、変換を実施する間のデータオーバーフローを制限し、メモリアクセスを減らす必要がある。具体的に言うと、YangのシステムのKLTは、計算的に複雑である。その一方で、変換サイズを減らすことによって、潜在的に、より大きい変換と比較したコーディング利得も減る。

【0052】

第6に、マルチチャンネル変換を指定する情報を送ることが、ビットレートに関して高コ

10

20

30

40

50

ストになる可能性がある。これは、YangのシステムのKLTに関して特にそうである。というのは、送られる共分散行列の変換係数が、実数であるからである。

【0053】

第7に、低ビットレートマルチチャネルオーディオに関して、再構成されるチャネルの品質が、非常に限られる。低ビットレートのコーディングの要件のほかに、これは、部分的に、情報が実際にエンコードされるチャネルの数をシステムが選択的に優雅に削減する能力がないことに起因する。

【0054】

3. 量子化および重みづけの非効率性

図1のエンコーダ(100)では、ウェイタ(140)が、オーディオデータの帯域にまたがるひずみを整形し、クオンタイザ(150)が、量子化ステップサイズをセットして、フレームに関するひずみの振幅を変更し、これによって品質とビットレートのバランスをとる。エンコーダ(100)は、ほとんどの応用例で品質とビットレートのよいバランスを達成するが、エンコーダ(100)は、まだ複数の短所を有する。

【0055】

第1に、エンコーダ(100)には、チャンネルレベルでの品質に対する直接制御が欠けている。重みづけ係数によって、個々のチャンネルの量子化帯域にまたがる全体的なひずみが整形される。この均一のスカラ量子化ステップサイズは、あるフレームのすべての周波数帯域およびチャンネルにまたがるひずみの振幅に影響する。すべてのチャンネルでの非常に高い品質または非常に低い品質の強制がないので、エンコーダ(100)には、すべてのチャンネルの再構成された出力の同等の品質または少なくとも匹敵する品質の設定に対する直接制御が欠けている。

【0056】

第2に、重みづけ係数がロッシイ圧縮されるので、エンコーダ(100)には、重みづけ係数の量子化の分解能に対する制御が欠けている。量子化行列の直接圧縮に関して、エンコーダ(100)は、量子化行列の要素を均一に量子化し、その後、差分コーディングおよびハフマンコーディングを使用する。マスク要素の均一の量子化は、使用可能なビットレートまたは信号の複雑さの変化に適応しない。その結果、量子化行列が、再構成されたオーディオの全体的に低い品質に対して必要以上に高い分解能でエンコードされる場合があり、量子化行列が、再構成されたオーディオの高い品質に対して使用すべき分解能より低い分解能でエンコードされる場合がある。

【0057】

第3に、エンコーダ(100)での量子化行列の直接圧縮では、量子化行列の時間的冗長性を活用することができない。直接圧縮では、特定の量子化行列内の冗長性が除去されるが、一連の量子化行列の時間的冗長性が無視される。

【0058】

C. オーディオチャンネルのダウンミキシング(down-mixing)

マルチチャネルオーディオのエンコーディングおよびデコーディングはさておき、Dolby Pro-Logicおよび複数の他のシステムは、マルチチャネルオーディオのダウンミキシングを実行して、異なる数のスピーカを有するスピーカ構成との互換性を容易にする。Dolby Pro-Logicのダウンミキシングでは、たとえば、4チャンネルが、2チャンネルにミックスダウンされ、2チャンネルのそれぞれが、元の4つのチャンネルのオーディオデータのある組合せを有する。この2チャンネルを、ステレオチャンネル装置で出力することができ、あるいは、4チャンネルを、2チャンネルから再構成して、4チャンネル機器で出力することができる。

【0059】

この性質のダウンミキシングによって、互換性問題の一部が解決されるが、これは、あるセット構成、たとえば、4チャンネルから2チャンネルへのダウンミキシングに制限される。さらに、ミキシングの式が、事前に決定され、信号に適応するための経時的な変化が許容されない。

10

20

30

40

50

【課題を解決するための手段】**【0060】**

要約すると、詳細な説明をする本発明は、オーディオエンコーディングおよびオーディオデコーディングでの量子化および逆量子化の戦略を対象とする。たとえば、オーディオエンコーダは、1つまたは複数の量子化（たとえば重みづけ）技法を使用して、オーディオデータの品質および/またはビットレートを改善する。これによって、全体的な聴取経験が改善され、コンピュータシステムが、高品質オーディオの作成、配信、および再生のより説得力のあるプラットフォームになる。本明細書で説明する本発明には、さまざまな技法およびツールが含まれ、これらは、組み合わせるまたは独立に使用することができる。

10

【0061】

本明細書で説明する本発明の第1の態様によれば、オーディオエンコーダは、複数のチャンネルのオーディオデータを量子化し、複数のチャンネルに関する複数のチャンネル固有量子化係数を適用する。たとえば、チャンネル固有量子化係数は、クオンタイザステップ変更子であり、これによって、エンコーダに、チャンネルの間での再構成品質のバランスに対するより多くの制御が与えられる。

【0062】

本明細書で説明する本発明の第2の態様によれば、オーディオエンコーダは、オーディオデータを量子化し、複数の量子化行列を適用する。エンコーダは、量子化行列の分解能を変更する。これによって、たとえば、エンコーダが、全体的な品質がよい場合により高い分解能を使用し、全体的な品質が低い場合により低い分解能を使用するように、量子化行列の要素の分解能を変更できるようになる。

20

【0063】

本明細書で説明する本発明の第3の態様によれば、オーディオエンコーダは、時間予測を使用して1つまたは複数の量子化行列を圧縮する。たとえば、エンコーダは、別の行列に関する現在の行列の予測を計算し、その後、現在の行列および予測から残差を計算する。この形で、エンコーダは、量子化行列に関連するビットレートを減らす。

【0064】

オーディオエンコーダに関して上で説明した態様のいくつかに関して、オーディオデコーダによって、対応する逆処理およびデコーディングが実行される。

30

【0065】

本発明のさまざまな特徴および効果は、添付図面に関して進行する、実施形態の以下の詳細な説明から明白になる。

【発明を実施するための最良の形態】**【0066】**

本発明の、説明される実施形態は、エンコーディングおよびデコーディングでオーディオ情報を処理する手法およびツールを対象とする。説明される実施形態では、オーディオエンコーダで、エンコーディング中に、複数の手法を使用してオーディオを処理する。オーディオデコーダでは、デコード中に、複数の手法を使用して、オーディオを処理する。本明細書のところどころで、単一の統合されたシステムの一部として手法を説明するが、これらの手法は、別々に、潜在的には他の手法と組み合わせ、適用することができる。代替実施形態では、エンコーダまたはデコーダ以外のオーディオ処理ツールによって、手法の1つまたは複数が実施される。

40

【0067】

いくつかの実施形態で、エンコーダが、マルチチャンネル前処理を実行する。低ビットレートコーディングについて、たとえば、エンコーダは、任意選択として、時間領域オーディオサンプルを再行列化して、相互チャンネル相関性を人工的に増やす。これによって、コーディングの複雑さを減らすことによって、影響されるチャンネルの後続の比較がより効率的になる。前処理によって、チャンネルセパレーションが低下するが、全体的な品質を改善することができる。

50

【 0 0 6 8 】

いくつかの実施形態で、エンコーダおよびデコーダが、ウィンドウのタイルに構成されたマルチチャンネルオーディオを扱う。たとえば、エンコーダが、チャンネルごとの基準でマルチチャンネルオーディオのフレームを区分し、各チャンネルが、他のチャンネルと独立のウィンドウ構成を有することができるようにする。エンコーダは、区分されたチャンネルのウィンドウを、マルチチャンネル変換用のタイルにグループ化する。これによって、エンコーダが、小さいウィンドウを有するフレームの特定のチャンネルに現れる推移を分離する（プリエコーアーチファクトを減らす）ことができるが、フレームの他のチャンネルでの周波数分解能および時間的冗長性削減に大きいウィンドウを使用することができるようになる。

【 0 0 6 9 】

いくつかの実施形態で、エンコーダが、1つまたは複数の柔軟なマルチチャンネル変換手法を実行する。デコーダは、対応する逆マルチチャンネル変換手法を実行する。第1の手法では、エンコーダが、エンコーダでの知覚的重みづけの後にマルチチャンネル変換を実行し、これによって、再構成時のチャンネルにまたがる可聴量子化雑音の漏れが減る。第2の手法では、エンコーダが、マルチチャンネル変換についてチャンネルを柔軟にグループ化して、異なる時にチャンネルを選択的に含める。第3の手法では、エンコーダが、柔軟にマルチチャンネル変換に特定の周波数帯域を含めるか除外して、互換性のある帯域を選択的に含める。第4の手法では、エンコーダが、選択的に事前定義の行列を使用するか、ギブンス回転を使用してカスタム変換行列をパラメータ化することによって、変化行列に関連するビットレートを減らす。第5の手法では、エンコーダが、柔軟な階層マルチチャンネル変換を実行する。

【 0 0 7 0 】

いくつかの実施形態で、エンコーダが、1つまたは複数の改善された量子化手法または改善された重みづけ手法を実行する。対応するデコーダが、対応する逆量子化手法または逆重みづけ手法を実行する。第1の手法では、エンコーダが、チャンネルごとの量子化ステップ変更子を計算し、適用し、この変更子によって、エンコーダに、チャンネル間の再構成品質のバランスに対するより多くの制御が与えられる。第2の手法では、エンコーダが、量子化行列要素の柔軟な量子化ステップサイズを使用し、これによって、エンコーダが、量子化行列要素の分解能を変更できるようになる。第3の手法では、エンコーダが、量子化行列の圧縮で時間予測を使用して、ビットレートを減らす。

【 0 0 7 1 】

いくつかの実施形態で、デコーダが、マルチチャンネル後処理を実行する。たとえば、デコーダが、任意選択として、時間領域オーディオサンプルを再行列化して、再生時にファントムチャンネルを作成し、特殊効果を実行し、より少ないスピーカでの再生のためまたは他の目的のためにチャンネルを折り畳む。

【 0 0 7 2 】

説明される実施形態では、マルチチャンネルオーディオに、図4の行列(400)に示されているように、標準的な5.1チャンネル/スピーカ構成の6チャンネルが含まれる。「5」チャンネルは、左、右、中央、左後ろ、および右後ろのチャンネルであり、サラウンド用に普通に空間的に配置される。「1」チャンネルは、サブウーファまたは低周波数効果チャンネルである。説明を明瞭にするために、行列(400)に示されたチャンネルの順序を、本明細書の残りの行列および式にも使用する。代替実施形態では、チャンネルの異なる順序付け、異なる数(たとえば7.1、9.1、2)、および/または構成を有するマルチチャンネルオーディオが使用される。

【 0 0 7 3 】

説明される実施形態で、オーディオエンコーダおよびオーディオデコーダは、さまざまな手法を実行する。これらの手法の動作を、提示のために通常は特定のシーケンシャルな順序で説明するが、この説明の形に、特定の順序付けが必要でない場合に、動作の順序の些細な再配置が含まれることを理解されたい。たとえば、順次説明される動作を、いくつかの場合に、再配置するか並列に実行することができる。さらに、説明を単純にするため

10

20

30

40

50

に、流れ図では、通常は、特定の手法を他の手法と共に使用することができるさまざまな形を示さない。

【0074】

I. コンピューティング環境

図5に、説明される実施形態を実施することができる適当なコンピューティング環境(500)の一般化された例を示す。コンピューティング環境(500)は、本発明の使用または機能性の範囲に関する制限を提案することを意図されたものではない。というのは、本発明を、別個の汎用コンピューティング環境または特殊目的コンピューティング環境で実施することができるからである。

【0075】

図5を参照すると、コンピューティング環境(500)に、少なくとも1つの処理ユニット(510)とメモリ(520)が含まれる。図5では、この最も基本的な構成(530)が、破線の中に含まれる。処理ユニット(510)は、コンピュータ実行可能命令を実行し、実際のプロセッサまたは仮想プロセッサとすることができる。マルチプロセッシングシステムでは、複数の処理ユニットが、コンピュータ実行可能命令を実行して、処理能力が増やされる。メモリ(520)は、揮発性メモリ(たとえば、レジスタ、キャッシュ、RAM)、不揮発性メモリ(たとえば、ROM、EEPROM、フラッシュメモリなど)、またはこの2つの組合せとすることができる。メモリ(520)には、説明される実施形態の1つまたは複数によるオーディオ処理手法を実施するソフトウェア(580)が保管される。

【0076】

コンピューティング環境が、追加の特徴を有することができる。たとえば、コンピューティング環境(500)に、ストレージ(540)、1つまたは複数の入力デバイス(550)、1つまたは複数の出力デバイス(560)、および1つまたは複数の通信接続(570)が含まれる。バス、コントローラ、またはネットワークなどの相互接続機構(図示せず)によって、コンピューティング環境(500)のコンポーネントが相互接続される。通常、オペレーティングシステムソフトウェア(図示せず)によって、コンピューティング環境(500)で実行される他のソフトウェアのオペレーティング環境が提供され、コンピューティング環境(500)のコンポーネントのアクティビティが調整される。

【0077】

ストレージ(540)は、取外し可能または取外し不能とすることができ、ストレージ(540)に、磁気ディスク、磁気テープ、磁気カセット、CD-ROM、CD-RW、DVD、または、情報を保管するのに使用でき、コンピューティング環境(500)内でアクセスできる他のメディアが含まれる。ストレージ(540)には、説明される実施形態の1つまたは複数によるオーディオ処理手法を実施するソフトウェア(580)の命令が保管される。

【0078】

入力デバイス(550)は、キーボード、マウス、ペン、またはトラックボールなどの接触入力デバイス、音声入力デバイス、スキャニングデバイス、ネットワークアダプタ、または、コンピューティング環境(500)に入力を供給する別のデバイスとすることができる。オーディオに関して、入力デバイス(550)を、アナログ形式またはデジタル形式のオーディオ入力を受け入れるサウンドカードまたは類似するデバイス、またはコンピューティング環境にオーディオサンプルを提供するCD-ROM/DVDリーダーとすることができる。出力デバイス(560)は、ディスプレイ、プリンタ、スピーカ、CD/DVDライター、ネットワークアダプタ、または、コンピューティング環境(500)から出力を供給する別のデバイスとすることができる。

【0079】

通信接続(570)によって、別のコンピューティングエンティティへの通信メディアを介する通信が可能になる。通信メディアは、コンピュータ実行可能命令、圧縮オーディオ情報、または変調されたデータ信号内の他のデータなどの情報を伝える。変調されたデ

10

20

30

40

50

ータ信号とは、情報を信号内でエンコードする形でその特性の1つまたは複数を設定されるか変更された信号である。制限ではなく例として、通信メディアに、電気、光、RF、赤外線、音響、または他の搬送波を用いて実施される有線または無線の手法が含まれる。

【0080】

本発明を、コンピュータ可読メディアの全般的な文脈で説明することができる。コンピュータ可読メディアとは、コンピュータ環境内でアクセスできるすべての使用可能なメディアである。制限ではなく例として、コンピューティング環境(500)に関して、コンピュータ可読メディアに、メモリ(520)、ストレージ(540)、通信メディア、およびこれらの任意の組合せが含まれる。

【0081】

本発明を、プログラムモジュールに含まれるものなど、コンピューティング環境内でターゲットの実際のプロセッサまたは仮想プロセッサ上で実行される、コンピュータ実行可能命令の全般的な文脈で説明することができる。一般に、プログラムモジュールには、特定のタスクを実行するか特定の抽象データ型を実施する、ルーチン、プログラム、ライブラリ、オブジェクト、クラス、コンポーネント、データ構造などが含まれる。プログラムモジュールの機能性を、さまざまな実施形態で、望み通りにプログラムモジュールの間で組み合わせるか分割することができる。プログラムモジュールのコンピュータ実行可能命令は、ローカルコンピューティング環境または分散コンピューティング環境内で実行することができる。

【0082】

提示のために、この詳細な説明では、「決定」、「生成」、「調節」、および「適用」などの単語を使用して、コンピューティング環境でのコンピュータ動作を説明する。これらの単語は、コンピュータによって実行される動作の高水準の抽象化であり、人間によって実行される動作と混同してはならない。これらの単語に対応する実際のコンピュータ動作は、実施形態に応じて変化する。

【0083】

II. 一般化されたオーディオエンコーダおよびオーディオデコーダ

図6は、説明される実施形態を実施することができる一般化されたオーディオエンコーダ(600)のブロック図である。図7は、説明される実施形態を実施することができる一般化されたオーディオデコーダ(700)のブロック図である。

【0084】

エンコーダおよびデコーダの中のモジュールの間に示された関係によって、エンコーダとデコーダでの情報の流れが示され、他の関係は、図を単純にするために示されていない。所望の圧縮の実施形態およびタイプに応じて、エンコーダまたはデコーダのモジュールを、追加し、省略し、複数のモジュールに分割し、他のモジュールと組み合わせ、かつ/または類似するモジュールと置換することができる。代替実施形態では、異なるモジュールおよび/または他の構成を有するエンコーダまたはデコーダによって、オーディオデータを処理する。

【0085】

A. 一般化されたオーディオエンコーダ

一般化されたオーディオエンコーダ(600)には、セレクタ(608)、マルチチャネルプリプロセッサ(610)、パーティショナ(分配器)(partitioner)/タイルコンフィギュアラ(configurer)(620)、周波数トランスフォーマ(630)、知覚モデラ(640)、量子化帯域ウェイタ(642)、チャンネルウェイタ(644)、マルチチャネルトランスフォーマ(650)、クオンタイザ(660)、エン트로ピエンコーダ(670)、コントローラ(680)、ミックスド/ピュアロスレスコーダ(672)および関連エン트로ピエンコーダ(674)、およびビットストリームマルチプレクサ[「MUX」](690)が含まれる。

【0086】

エンコーダ(600)は、あるサンプリング深さとサンプリングレートの入力オーディ

10

20

30

40

50

オサンプル(605)の時系列を、パルスコード変調[「PCM」]フォーマットで受け取る。説明される実施形態のほとんどについて、入力オーディオサンプル(605)は、マルチチャンネルオーディオ(たとえば、ステレオ、サラウンド)用であるが、入力オーディオサンプル(605)を、その代わりにモノラルとすることができる。エンコーダ(600)は、オーディオサンプル(605)を圧縮し、エンコーダ(600)のさまざまなモジュールによって作られる情報を多重化して、Windows(登録商標)Media Audio[「WMA」]フォーマットまたはAdvanced Streaming Format[「ASF」]などのフォーマットでビットストリーム(695)を出力する。その代わりに、エンコーダ(600)が、他の入力フォーマットおよび/または出力フォーマットを扱うことができる。

10

【0087】

セレクタ(608)は、オーディオサンプル(605)に関する複数のエンコーディングモードの間で選択する。図6では、セレクタ(608)が、ミックスド/ピュアロスレスコーディングモードとロッシイコーディングモードの間で切り替える。ロスレスコーディングモードには、ミックスド/ピュアロスレスコーダ(672)が含まれ、ロスレスコーディングモードは、通常は、高品質(および高ビットレート)の圧縮に使用される。ロッシイコーディングモードには、ウェイタ(642)およびクオンタイザ(660)などのコンポーネントが含まれ、ロッシイコーディングモードは、通常は調整可能な品質(および制御されたビットレート)の圧縮に使用される。セレクタ(608)での選択判断は、ユーザ入力または他の判断基準に依存する。ある状況(たとえば、ロッシイ圧縮で適当な品質を配信できないか、ビットが過剰に作られる時)では、エンコーダ(600)が、あるフレームまたはフレームの組について、ロッシイコーディングからミックスド/ピュアロスレスコーディングに切り替えることができる。

20

【0088】

マルチチャンネルオーディオデータのロッシイコーディングについて、マルチチャンネルプリプロセッサ(610)は、任意選択として、時間領域オーディオサンプル(605)を再行列化する。いくつかの実施形態で、マルチチャンネルプリプロセッサ(610)は、オーディオサンプル(605)を選択的に再行列化して、1つまたは複数のコーディングされたチャンネルを捨てるか、エンコーダ(600)内の相互チャンネル相関性を増やすが、それでもデコーダ(700)での再構成(ある形での)を可能にする。これによって、エンコーダに、チャンネルレベルでの品質に対する追加の制御が与えられる。マルチチャンネルプリプロセッサ(610)は、マルチチャンネル後処理の命令などのサイド情報を、MUX(690)に送ることができる。いくつかの実施形態でのマルチチャンネルプリプロセッサの動作に関する追加の詳細については、「マルチチャンネル前処理」という題名のセクションを参照されたい。代替案では、エンコーダ(600)が、別の形のマルチチャンネル前処理を実行する。

30

【0089】

パーティショナ/タイルコンフィギュアラ(620)は、オーディオ入力サンプル(605)のフレームを、時間依存性サイズ関数およびウィンドウ整形関数(time-varying size and window shaping functions)を有するサブフレームブロック(すなわちウィンドウ)に区分する。サブフレームブロックのサイズおよびウィンドウは、フレーム内の推移信号の検出、コーディングモード、ならびに他の要因に依存する。

40

【0090】

エンコーダ(600)が、ロッシイコーディングからミックスド/ピュアロスレスコーディングに切り替える場合に、サブフレームブロックは、理論的にはオーバーラップする必要も、ウィンドウウィング関数を有する必要もない(すなわち、オーバーラップしない長方形のウィンドウブロック)が、ロッシイコーディングされたフレームと他のフレームの間の推移は、特別な扱いを必要とする可能性がある。パーティショナ/タイルコンフィギュアラ(620)は、区分されたデータのブロックを、ミックスド/ピュアロスレスコ

50

ーダ(672)に出力し、ブロックサイズなどのサイド情報をMUX(690)に出力する。ミックスまたはピュアのロスレスコーディングされたフレームの区分およびウィンドウイングに関する追加の詳細については、関連特許出願の発明の名称"Unified Lossy and Lossless Audio Compression"の米国特許出願第60/408432号を参照されたい。

【0091】

エンコーダ(600)が、ロッシイコーディングを使用する時には、可変サイズウィンドウによって、可変時間分解能が可能になる。小さいブロックを用いると、短いアクティブな推移セグメントで、時間詳細のより多くの保存が可能になる。大きいブロックは、よりよい周波数分解能とより悪い時間分解能を有し、通常は、大きいブロックによって、より長くより少数のアクティブセグメントでのより高い圧縮効率が可能になる。これは、部分的にはフレームヘッダおよびサイド情報が、小さいブロックよりもサイズに比例して少なくなるからであり、部分的にはこれによってよりよい冗長性削減が可能になるからである。ブロックをオーバーラップさせて、そうでなければ後の量子化によって導入される可能性があるブロック間の知覚可能な不連続性を減らすことができる。パーティショナ/タイルコンフィギュアラ(620)は、区分されたデータのブロックを周波数トランスフォーマ(630)に出力し、ブロックサイズなどのサイド情報をMUX(690)に出力する。いくつかの実施形態での推移検出および区分判断基準に関する追加情報については、参照によって本明細書に組み込まれる関連特許出願の発明の名称"Adaptive Window-Size Selection in Transform Coding,"の米国特許出願第10/016,918(2001年12月14日出願)を参照されたい。代替案では、パーティショナ/タイルコンフィギュアラ(620)で、フレームをウィンドウに区分する時に、他の区分判断基準またはブロックサイズを使用する。

【0092】

いくつかの実施形態で、パーティショナ/タイルコンフィギュアラ(620)は、マルチチャンネルオーディオのフレームをチャンネルごとに区分する。パーティショナ/タイルコンフィギュアラ(620)は、品質/ビットレートから許容される場合に、フレーム内の各チャンネルを独立に区分する。これによって、たとえば、パーティショナ/タイルコンフィギュアラ(620)が、より小さいウィンドウを用いて特定のチャンネルに現れる推移を分離するが、他のチャンネルで周波数分解能または圧縮効率のためにより大きいウィンドウを使用することが可能になる。これによって、チャンネルごとに推移を分離することによって圧縮効率を改善することができるが、多くの場合に、個々のチャンネル内の区分を指定する追加情報が、必要になる。時間的に同一位置にある同一サイズのウィンドウは、マルチチャンネル変換を介するさらなる冗長性削減の資格を有する場合がある。したがって、パーティショナ/タイルコンフィギュアラ(620)は、時間的に同一位置にある同一サイズのウィンドウを、タイルとしてグループ化する。いくつかの実施形態でのタイリングに関する追加の詳細については、「タイル構成」という題名のセクションを参照されたい。

【0093】

周波数トランスフォーマ(630)は、オーディオサンプルを受け取り、周波数領域のデータに変換する。周波数トランスフォーマ(630)は、周波数係数データのブロックをウェイト(642)に出力し、ブロックサイズなどのサイド情報をMUX(690)に出力する。周波数トランスフォーマ(630)は、周波数係数とサイド情報の両方を知覚モダ(640)に出力する。いくつかの実施形態で、周波数トランスフォーマ(630)は、時間に伴って変化する変調ラップド変換(Modulated Lapped Transform) [「MLT」] をサブフレームブロックに適用するが、このMLTは、サブフレームブロックの正弦ウィンドウ関数によって変調されたDCTに似た演算である。代替実施形態では、MLTの他の変形形態またはDCTあるいは、変調ありまたはなしの、オーバーラップありまたはなしの、他のタイプの周波数変換を使用するか、サブバンドコーディングまたはウェーブレットコーディングを使用する。

【0094】

知覚モデラ(640)によって、人間の聴覚系のプロパティをモデル化して、所与のビットレートの再構成されたオーディオ信号の知覚される品質を改善する。一般に、知覚モデラ(640)は、聴覚モデルに従ってオーディオデータを処理し、その情報をウェイト(642)に供給し、このウェイト(642)は、オーディオデータの重みづけ係数を生成するのに使用することができる。知覚モデラ(640)は、さまざまな聴覚モデルのいずれかを使用し、励起パターン情報または他の情報をウェイト(642)に渡す。

【0095】

量子化帯域ウェイト(642)は、知覚モデラ(640)から受け取った情報に基づいて量子化行列の重みづけ係数を生成し、その重みづけ係数を、周波数トランスフォーマ(630)から受け取ったデータに適用する。量子化行列の重みづけ係数には、オーディオデータの複数の量子化帯域のそれぞれの重みが含まれる。量子化帯域は、数またはエンコーダ(600)の他所で使用される臨界帯域からの位置において、同一または異なるものとしてことができ、重みづけ係数を、ブロックごとに、振幅および量子化帯域の数において変更することができる。量子化帯域ウェイト(642)は、係数データの重みづけされたブロックをチャンネルウェイト(644)に出力し、重みづけされた係数の組などのサイド情報をMUX(690)に出力する。重みづけされた係数の組を、さらに効率的な表現のために圧縮することができる。重みづけ係数が、ロッシイ圧縮される場合には、再構成される重みづけ係数が、通常は、係数データのブロックに重みをつけるのに使用される。いくつかの実施形態での重みづけ係数の計算および圧縮に関する追加の詳細については、「量子化および重みづけ」という題名のセクションを参照されたい。代替案では、エンコーダ(600)が、別の形の重みづけを使用するか、重みづけをスキップする。

【0096】

チャンネルウェイト(644)は、知覚モデラ(640)から受け取った情報およびローカルに再構成された信号の品質に基づいて、チャンネルのチャンネル固有重みづけ係数(スカラーである)を生成する。スカラー重み(量子化ステップ変更子とも称する)を用いると、エンコーダ(600)が、再構成されるチャンネルに、近似的に均一の品質を与えられるようになる。チャンネル重み係数は、チャンネルごとおよびブロックごとに、またはある他のレベルで、振幅を変えることができる。チャンネルウェイト(644)は、係数データの重みづけされたブロックをマルチチャンネルトランスフォーマ(650)に出力し、チャンネル重み係数の組などのサイド情報をMUX(690)に出力する。流れ図のチャンネルウェイト(644)および量子化帯域ウェイト(642)は、入れ替えるか、一緒に組み合わせることができる。いくつかの実施形態での重みづけ係数の計算および圧縮に関する追加の詳細については、「量子化および重みづけ」という題名のセクションを参照されたい。代替案では、エンコーダ(600)が、別の形の重みづけを使用するか、重みづけをスキップする。

【0097】

マルチチャンネルオーディオデータに関して、チャンネルウェイト(644)によって作られる雑音形の(noise-shaped)周波数係数データの複数のチャンネルが、しばしば相関し、したがって、マルチチャンネルトランスフォーマ(650)が、マルチチャンネル変換を適用することができる。たとえば、マルチチャンネルトランスフォーマ(650)は、タイルのチャンネルおよび/または量子化帯域のすべてではなく一部に、マルチチャンネル変換を選択的に柔軟に適用する。これによって、マルチチャンネルトランスフォーマ(650)に、タイルの比較的相関する部分への変換の適用に対する正確な制御が与えられる。計算的な複雑さを減らすために、マルチチャンネルトランスフォーマ(650)は、1レベル変換ではなく階層変換を使用することができる。変換行列に関連するビットレートを減らすために、マルチチャンネルトランスフォーマ(650)は、事前定義の行列(たとえば、恒等変換/無変換、アダマール、DCTタイプII)またはカスタム行列を選択的に使用し、カスタム行列に効率的な圧縮を適用する。最後に、マルチチャンネル変換は、ウェイト(642)の下流なので、デコーダ(700)での逆マルチチャンネル変換の後のチャンネル間で漏れる雑音を知覚できること(たとえば、後続の量子化に起因する)が、逆重み

10

20

30

40

50

づけによって制御される。いくつかの実施形態でのマルチチャンネル変換に関する追加の詳細については、「柔軟なマルチチャンネル変換」という題名のセクションを参照されたい。代替案では、エンコーダ(600)が、他の形のマルチチャンネル変換を使用するか、まったく変換を行わない。マルチチャンネルトランスフォーマ(650)は、MUX(690)へのサイド情報を作って、たとえば、使用されたマルチチャンネル変換およびタイルのマルチチャンネル変換された部分を示す。

【0098】

クオンタイザ(660)は、マルチチャンネルトランスフォーマ(650)の出力を量子化し、エントロピエンコーダ(670)への量子化された係数データおよびMUX(690)への量子化ステップサイズを含むサイド情報を作る。図6では、クオンタイザ(660)が、タイルごとに量子化係数を計算する適応式均一スカラクオンタイザである。タイル量子化係数を、量子化ループの反復ごとに変更して、エントロピエンコーダ(670)出力のビットレートに影響を及ぼすことができ、チャンネルごとの量子化ステップ変更子を使用して、チャンネルの間の再構成品質のバランスをとることができる。いくつかの実施形態での量子化に関する追加の詳細については、「量子化および重みづけ」という題名のセクションを参照されたい。代替実施形態では、クオンタイザが、不均一クオンタイザ、ベクトルクオンタイザ、および/または非適応クオンタイザであるか、異なる形の適応均一スカラ量子化を使用する。他の代替実施形態では、クオンタイザ(660)、量子化帯域ウェイト(642)、チャンネルウェイト(644)、およびマルチチャンネルトランスフォーマ(650)が、融合され、融合されたモジュールが、さまざまな重みをすべて一緒に判定する。

【0099】

エントロピエンコーダ(670)は、クオンタイザ(660)から受け取った量子化された係数データをロスレス圧縮する。いくつかの実施形態で、エントロピエンコーダ(670)は、関連特許出願の発明の名称"Entropy Coding by Adapting Coding Between Level and Run Length/Level Modes"の米国特許出願第60/408,538号に記載の適応エントロピコーディングを使用する。代替案では、エントロピエンコーダ(670)が、マルチレベルランレングスコーディング、可変長対可変長コーディング、ランレングスコーディング、ハフマンコーディング、辞書コーディング、算術コーディング、LZコーディング、または他のエントロピコーディング手法の他の形または組合せを使用する。エントロピエンコーダ(670)は、オーディオ情報のエンコーディングに費やされるビット数を計算し、この情報をレート/品質コントローラ(680)に渡すことができる。

【0100】

コントローラ(680)は、クオンタイザ(660)と共に働いて、エンコーダ(600)の出力のビットレートおよび/または品質を調整する。コントローラ(680)は、エンコーダ(600)の他のモジュールから情報を受け取り、受け取った情報を処理して、現在の条件に対して所望の量子化係数を判定する。コントローラ(680)は、品質制約および/またはビットレート制約を満足するという目標をもって、クオンタイザ(660)に量子化係数を出力する。

【0101】

ミックスド/ピュアロスレスコーダ(672)および関連エントロピエンコーダ(674)は、ミックスド/ピュアロスレスコーディングモードでオーディオデータを圧縮する。エンコーダ(600)は、シーケンス全体にミックスド/ピュアロスレスコーディングモードを使用するか、フレームごと、ブロックごと、タイルごと、または他の基準でコーディングモードを切り替える。ミックスド/ピュアロスレスコーディングモードに関する追加の詳細については、関連特許出願の発明の名称"Unified Lossy and Lossless Audio Compression"の米国特許出願第60/408432号を参照されたい。代替案では、エンコーダ(600)が、ミックスドおよび/またはピュアのロスレスエンコーディングの他の手法を使用する。

【0102】

10

20

30

40

50

MUX(690)は、オーディオエンコーダ(600)の他のモジュールから受け取ったサイド情報を、エントロピエンコーダ(670、674)から受け取ったエントロピエンコーディングされたデータと多重化する。MUX(690)は、WMAフォーマットまたはオーディオデコーダが認識する別のフォーマットで情報を出力する。MUX(690)には、エンコーダ(600)によって出力されるビットストリーム(695)を保管する仮想バッファが含まれる。仮想バッファは、比較的一定のビットレートでデータを出力し、品質は、入力複雑さの変化に起因して変化する可能性がある。バッファの現在の満杯度および他の特性を、コントローラ(680)によって使用して、品質および/またはビットレートを調整することができる。代替案では、出力ビットレートが、経時的に変化することができ、品質が、比較的一定に保たれる。あるいは、出力ビットレートが、特定のビットレート未満に制限されるだけであり、このビットレートは、一定にまたは時間的に変換するのいずれかである。

10

【0103】

B. 一般化されたオーディオデコーダ

図7を参照すると、一般化されたオーディオデコーダ(700)に、ビットストリームデマルチプレクサ[「DEMUX」](710)、1つまたは複数のエントロピデコーダ(720)、ミックスド/ピュアロスレスデコーダ(722)、タイル構成デコーダ(730)、逆マルチチャネルトランスフォーマ(740)、逆クオンタイザ/ウェイト(750)、逆周波数トランスフォーマ(760)、オーバーラップ/アダー(770)、およびマルチチャネルポストプロセッサ(780)が含まれる。デコーダ(700)にはレ

20

【0104】

デコーダ(700)は、WMAフォーマットまたは別のフォーマットの圧縮オーディオ情報のビットストリーム(705)を受け取る。ビットストリーム(705)には、エントロピエンコーディングされたデータならびにサイド情報が含まれ、デコーダ(700)は、それらからオーディオサンプル(795)を再構成する。

【0105】

DEMUX(710)は、ビットストリーム(705)の情報を解析し、情報をデコーダ(700)のモジュールに送る。DEMUX(710)には、オーディオの複雑さの変動、ネットワークジッタ、および/または他の要因に起因するビットレートの短期間変動を補償するために、1つまたは複数のバッファが含まれる。

30

【0106】

1つまたは複数のエントロピデコーダ(720)は、DEMUX(710)から受け取るエントロピコードをロスレス圧縮解除する。エントロピデコーダ(720)は、通常は、エンコーダ(600)で使用されるエントロピエンコード手法の逆を適用する。説明を単純にするために、1つのエントロピデコーダモジュールを図7に示したが、異なるエントロピデコーダを、ロッシイコーディングモードとロスレスコーディングモードに使用することができ、1つのモードの中で異なるエントロピデコーダを使用することもできる。また、説明を単純にするために、図7には、モード選択論理が示されていない。ロッシイコーディングモードで圧縮されたデータをデコードする時に、エントロピデコーダ(720)は、量子化された周波数係数データを作る。

40

【0107】

ミックスド/ピュアロスレスデコーダ(722)および関連するエントロピデコーダ(720)は、ミックスド/ピュアロスレスコーディングモードのロスレスエンコーディングされたオーディオデータを圧縮解除する。ミックスド/ピュアロスレスデコーディングモードの圧縮解除に関する追加の詳細については、関連特許出願の発明の名称"Unified Lossy and Lossless Audio Compression"の米国特許出願第60/408432号を参照されたい。代替案では、デコーダ(700)が、ミックスドおよび/またはピュアのロスレスデコーディングの他の手法を使用する。

50

【 0 1 0 8 】

タイル構成デコーダ(730)は、DEMUX(710)から、フレームのタイルのパターンを示す情報を受け取り、必要な場合にデコードする。タイルパターン情報は、エン트로ピエンコーディングされるか、他の形でパラメータ化される可能性がある。タイル構成デコーダ(730)は、タイルパターン情報を、デコーダ(700)のさまざまな他のモジュールに渡す。いくつかの実施形態でのタイル構成デコーディングに関する追加の詳細については、「タイル構成」という題名のセクションを参照されたい。代替案では、デコーダ(700)が、フレーム内のウィンドウパターンをパラメータ化する他の手法を使用する。

【 0 1 0 9 】

逆マルチチャネルトランスフォーマ(740)は、エン트로ピデコーダ(720)からの量子化された周波数係数データならびにタイル構成デコーダ(730)からのタイルパターン情報および、たとえば使用されたマルチチャネル変換およびタイルの変換された部分を示す、DEMUX(710)からのサイド情報を受け取る。この情報を使用して、逆マルチチャネルトランスフォーマ(740)は、必要に応じて変換行列を圧縮解除し、1つまたは複数の逆マルチチャネル変換をオーディオデータに選択的に柔軟に適用する。逆クオンタイザ/ウェイタ(750)に間する逆マルチチャネルトランスフォーマ(740)の配置は、チャンネルにまたがって漏れる可能性がある量子化雑音を整形するのに役立つ。いくつかの実施形態の逆マルチチャネルトランスフォーマに関する追加の詳細については、「柔軟なマルチチャネル変換」という題名のセクションを参照されたい。

【 0 1 1 0 】

逆クオンタイザ/ウェイタ(750)は、タイルおよびチャンネルの量子化係数ならびに量子化行列をDEMUX(710)から受け取り、量子化された周波数係数データを逆マルチチャネルトランスフォーマ(740)から受け取る。逆クオンタイザ/ウェイタ(750)は、受け取った量子化係数/行列情報を必要に応じて圧縮解除し、逆量子化および重みづけを実行する。いくつかの実施形態での逆量子化および重みづけの追加の詳細については、「量子化および重みづけ」という題名のセクションを参照されたい。代替実施形態では、逆クオンタイザ/ウェイタによって、エンコーダで使用される他の量子化手法の逆が適用される。

【 0 1 1 1 】

逆周波数トランスフォーマ(760)は、逆クオンタイザ/ウェイタ(750)によって出力される周波数係数データならびにDEMUX(710)からのサイド情報およびタイル構成デコーダ(730)からのタイルパターン情報を受け取る。逆周波数トランスフォーマ(760)は、エンコーダで使用される周波数変換の逆を適用し、ブロックをオーバーラップ/アダー(770)に出力する。

【 0 1 1 2 】

タイル構成デコーダ(730)からタイルパターン情報を受け取るほかに、オーバーラップ/アダー(770)は、逆周波数トランスフォーマ(760)および/またはミックスド/ピュアロスレスデコーダ(722)からデコードされた情報も受け取る。オーバーラップ/アダー(770)は、必要に応じてオーディオデータをオーバーラップさせ、加算し、異なるモードでエンコードされたオーディオデータのフレームまたは他のシーケンスをインターリーブする。ミックスドまたはピュアのロスレスコーディングされたフレームのオーバーラップ、加算、およびインターリーブに関する追加の詳細は、関連特許出願の発明の名称 "Unified Lossy and Lossless Audio Compression" の米国特許出願第 60 / 408432号を参照されたい。代替案では、デコーダ(700)が、フレームのオーバーラップ、加算、およびインターリーブに他の手法を使用する。

【 0 1 1 3 】

マルチチャネルポストプロセッサ(780)は、任意選択として、オーバーラップ/アダー(770)によって出力される時間領域オーディオサンプルを再行列化する。マルチチャネルポストプロセッサは、オーディオデータを選択的に再行列化して、再生用のファ

10

20

30

40

50

ントムチャンネルを作成し、スピーカの間でのチャンネルの空間的回転、より少数のスピーカでの再生または他の目的のためのチャンネルの折り曲げなどの特殊効果を実行する。ビットストリーム制御された後処理について、後処理変換行列は、経時的に変化し、シグナリングされるかビットストリーム(705)に含まれる。いくつかの実施形態でのマルチチャンネルポストプロセッサの動作に関する追加の詳細は、「マルチチャンネル後処理」という題名のセクションを参照されたい。代替案では、デコーダ(700)が、別の形のマルチチャンネル後処理を実行する。

【0114】

III. マルチチャンネル前処理

いくつかの実施形態で、図6のエンコーダ(600)などのエンコーダが、時間領域の入力オーディオサンプルに対してマルチチャンネル前処理を実行する。

【0115】

一般に、入力としてN個のソースオーディオチャンネルがある時に、エンコーダによって作られるコーディングされたチャンネルの数もNになる。コーディングされたチャンネルが、ソースチャンネルと1対1対応する場合があります、あるいは、コーディングされたチャンネルが、マルチチャンネル変換コーディングされたチャンネルである場合がある。しかし、ソースのコーディングの複雑さによって圧縮が困難になる時、またはエンコーダバッファが満杯である時には、エンコーダが、元の入力オーディオチャンネルの1つまたは複数を変更するか捨てる(すなわちコーディングしない)場合がある。これは、コーディングの複雑さを減らし、オーディオの全体的な知覚される品質を改善するために行うことができる。品質駆動の前処理について、エンコーダは、測定されたオーディオ品質に反応してマルチチャンネル前処理を実行して、全体的なオーディオ品質およびチャンネルセパレーションを滑らかに制御する。

【0116】

たとえば、エンコーダは、マルチチャンネルオーディオイメージを変更して、1つまたは複数のチャンネルをよりクリティカルでないようにすることができ、その結果、チャンネルがエンコーダで捨てられるが、デコーダで「ファントムチャンネル」として再構成されるようになる。チャンネルの徹底的な削除は、品質に劇的に影響する可能性があり、したがって、これは、コーディングの複雑さが非常に高いか、バッファが非常に満杯であり、他の手段を介して良い品質の再生を達成できない時に限って行われる。

【0117】

エンコーダは、コーディングされるチャンネルの数が、出力のチャンネル数より少ない時にどの処置を講ずるかをデコーダに示すことができる。その後、マルチチャンネル後処理変換をデコーダで使用して、下の「マルチチャンネル後処理」という題名のセクションで説明するように、ファントムチャンネルを作成することができる。あるいは、エンコーダが、別の目的のマルチチャンネル後処理を実行するようにデコーダに知らせることができる。

【0118】

図8に、マルチチャンネル前処理の一般化された手法(800)を示す。エンコーダが、時間領域マルチチャンネルオーディオデータ(805)に対するマルチチャンネル前処理を実行し(810)、時間領域の変換されたオーディオデータ(815)を作る。たとえば、前処理に、一般的なN対N変換が含まれ、このNは、チャンネルの数である。エンコーダは、N個のサンプルに行列Aをかける。

$$y_{pre} = A_{pre} \times x_{pre} \quad (4)$$

ここで、 x_{pre} および y_{pre} は、前処理に入力されるN個の入力および前処理から出力されるN個の出力であり、 A_{pre} は、実数(すなわち連続的な)値の要素を有する一般的な $N \times N$ 変換行列である。行列 A_{pre} は、 x_{pre} と比較して y_{pre} の相互チャンネル相関性を人工的に増やすように選択することができる。これによって、エンコーダの残りに関する複雑さが減るが、チャンネルセパレーションの低下が犠牲になる。

【0119】

出力 y_{pre} が、エンコーダの残りに供給され、これによって、図6に示された手法ま

10

20

30

40

50

たは他の圧縮手法を使用してデータがエンコードされ(820)、エンコードされたマルチチャンネルオーディオデータ(825)が作られる。

【0120】

エンコーダおよびデコーダによって使用される構文(syntax)によって、一般的なまたは事前定義の後処理マルチチャンネル変換行列の記述が可能になり、この後処理マルチチャンネル変換行列は、フレームごとに変更するか、オン/オフにすることができる。エンコーダは、この柔軟性を使用して、ステレオ/サラウンドイメージ減損を制限し、相互チャンネル相関性を人工的に増やすことによって、ある状況でチャンネルセパレーションとよりよい総合的な品質をトレードオフする。代替案では、デコーダおよびエンコーダが、マルチチャンネル前処理およびマルチチャンネル後処理の別の構文、たとえば、フレームごと以外の基礎での変換行列の変更を可能にする構文を使用する。

10

【0121】

図9aから図9eに、ある状況の下でエンコーダで相互チャンネル相関性を人工的に増やすのに使用されるマルチチャンネル前処理変換行列(900から904)を示す。エンコーダは、前処理行列の間で切り替えて、5.1チャンネル再生環境で、左チャンネル、右チャンネル、および中央チャンネルの間、および左後チャンネルと右後チャンネルの間で相互チャンネル相関性を人工的にどれほど増やすかを変更する。

【0122】

一実施形態で、低ビットレートで、エンコーダが、ある時間の期間にわたって再構成されたオーディオの品質を評価し、その結果に応じて、前処理行列の1つを選択する。エンコーダによって評価される品質測定は、雑音興奮比率[「NER」]であり、これは、元のデジタルオーディオクリップのエネルギーに対する再構成されたオーディオクリップの雑音パターンのエネルギーの比である。低いNER値は、良い品質を示し、高いNER値は、低い品質を示す。エンコーダは、1つまたは複数の前にエンコードされたフレームのNERを評価する。NERおよび他の品質測定に関する追加情報については、参照によって本明細書に組み込まれる関連特許出願の発明の名称"Techniques for Measurement of Perceptual Audio Quality,"の米国特許出願第10/017,861号(2001年12月14日出願)を参照されたい。代替案では、エンコーダが、別の品質測定、バッファ満杯度、および/またはある他の判断基準を使用して、前処理変換行列を選択し、あるいは、エンコーダが、マルチチャンネルオーディオの異なる期間を評価する。

20

30

【0123】

図9aから図9eに示された例に戻ると、低ビットレートで、エンコーダは、オーディオクリップの特定の範囲の NER_n に基づいて、前処理変換行列をゆっくり変更する。エンコーダは、 n の値を閾値 n_{low} および n_{high} と比較するが、これらの閾値は、実装依存である。一実施形態では、 n_{low} および n_{high} が、所定の値 $n_{low} = 0.05$ および $n_{high} = 0.1$ を有する。代替案では、 n_{low} および n_{high} が、ビットレートまたは他の判断基準に反応して経時的に変化する1つまたは複数の異なる値を有し、あるいは、エンコーダが、異なる数の行列の間で切り替える。

【0124】

低い値の n (たとえば、 n_{low})は、良い品質のコーディングを示す。したがって、エンコーダは、図9aに示された単位行列 A_{low} (900)を使用し、効果的に前処理をオフにする。

40

【0125】

その一方で、高い値の n (たとえば、 n_{high})は、低い品質のコーディングを示す。したがって、エンコーダは、図9cに示された行列 $A_{high,1}$ (902)を使用する。行列 $A_{high,1}$ (902)によって、激しいサラウンドイメージひずみが導入されるが、それと同時に、左チャンネル、右チャンネル、および中央チャンネルの間の非常に高い相関が押し付けられ、これによって、複雑さを減らすことによって後続のコーディング効率が改善される。マルチチャンネル変換された中央チャンネルは、元の左チャンネル、右チャンネル、および中央チャンネルの平均である。行列 $A_{high,1}$ (902)によって、後

50

チャンネルの間のチャンネルセパレーションも妥協して処理され、入力の左後ろチャンネルと右後ろチャンネルの平均がとられる。

【0126】

中間の値の n (たとえば、 $n_{low} < n < n_{high}$) は、中間の品質のコーディングを示す。したがって、エンコーダは、図9bに示された中間行列 $A_{inter,1}(901)$ を使用することができる。中間行列 $A_{inter,1}(901)$ では、係数 α によって、 n_{low} と n_{high} の間の n の相対位置が測定される。

【0127】

【数3】

$$\alpha = \frac{n - n_{low}}{n_{high} - n_{low}} \tag{5}$$

10

【0128】

中間行列 $A_{inter,1}(901)$ は、単位行列 $A_{low}(900)$ から低品質行列 $A_{high,1}(902)$ へ徐々に推移する。

【0129】

図9bおよび図9cに示された行列 $A_{inter,1}(901)$ および $A_{high,1}(902)$ について、エンコーダは、後に、エンコーダが相互チャンネル相関性を人工的に増やしたチャンネルの間の冗長性を活用し、エンコーダは、これらのチャンネルに関してマルチチャンネル後処理を実行するようにエンコーダに指示する必要がない。

20

【0130】

デコーダが、マルチチャンネル後処理を実行する能力を有する時には、エンコーダは、中央チャンネルの再構成をデコーダに委任することができる。そうである場合に、NER値 n によって、低い品質のコーディングが示される時に、エンコーダは、図9eに示された行列 $A_{high,2}(904)$ を使用するが、この行列を用いると、入力中央チャンネルが左右のチャンネルに漏れる。出力では、中央チャンネルが0であり、コーディングの複雑さが減る。

【0131】

【数4】

$$\begin{bmatrix} \left(\frac{a}{1.5} + \frac{.5 \cdot c}{1.5}\right) \\ \left(\frac{b}{1.5} + \frac{.5 \cdot c}{1.5}\right) \\ 0 \\ d \\ \frac{e+f}{2} \\ \frac{e+f}{2} \end{bmatrix} = A_{high,2} \cdot \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}$$

30

40

【0132】

エンコーダは、前処理変換行列 $A_{high,2}(904)$ を使用する時に、デコードされた左右のチャンネルの平均をとることによってファントム中央を作成するようにデコーダに(ビットストリームを介して)指示する。エンコーダでの後のマルチチャンネル変換では、平均をとられた後ろの左右のチャンネル(後処理なし)の間の冗長性を活用することができる、あるいは、エンコーダが、後ろの左右のチャンネルに関するあるマルチチャンネル後処理を実行するようにデコーダに指示することができる。

【0133】

NER値 n によって、中間の品質のコーディングが示される時には、エンコーダは、図9dに示された中間行列 $A_{inter,2}(903)$ を使用して、図9aおよび9eに示

50

された行列の間で推移することができる。

【 0 1 3 4 】

図 1 0 に、フレームごとに変換行列が潜在的に変化するマルチチャンネル前処理の手法 (1 0 0 0) を示す。変換行列の変更は、注意深く処理されない場合に、最終出力の可聴雑音 (たとえばポンという音) につながる可能性がある。ポンという雑音を導入しないようにするために、エンコーダは、ある変換行列から別の変換行列へ、フレームの間に徐々に推移する。

【 0 1 3 5 】

エンコーダは、まず、上で説明した前処理変換行列をセットする (1 0 1 0)。次に、エンコーダは、現在のフレームの行列が、前のフレーム (前のフレームがある場合に) の行列と異なるかどうかを判定する (1 0 2 0)。現在の行列が同一であるか、前の行列がない場合には、エンコーダは、現在のフレームの入力オーディオサンプルに行列を適用する (1 0 3 0)。そうでない場合には、エンコーダは、現在のフレームの入力オーディオサンプルにブレンドされた変換行列を適用する (1 0 4 0)。ブレンディング関数は、実施形態に依存する。一実施形態では、現在のフレームのサンプル i で、エンコーダが、短期間ブレンドされた行列 $A_{pre,i}$ を使用する。

【 0 1 3 6 】

【 数 5 】

$$A_{pre,i} = \frac{\text{NumSamples} - i}{\text{NumSamples}} A_{pre,prev} + \frac{i}{\text{NumSamples}} A_{pre,current} \quad (6)$$

【 0 1 3 7 】

ここで、 $A_{pre,prev}$ および $A_{pre,current}$ は、それぞれ前のフレームおよび現在のフレームの前処理行列であり、 NumSamples は、現在のフレームのサンプル数である。代替案では、エンコーダが、別のブレンディング関数を使用して、前処理変換行列の不連続性を平滑化する。

【 0 1 3 8 】

次に、エンコーダは、図 6 に示した手法または他の圧縮手法を使用して、フレームのマルチチャンネルオーディオデータをエンコードする (1 0 5 0)。エンコーダは、フレームごとに手法 (1 0 0 0) を繰り返す。代替案では、エンコーダが、他の基礎に基づいてマルチチャンネル前処理を変更する。

【 0 1 3 9 】

IV . タイル構成

いくつかの実施形態で、図 6 のエンコーダ (6 0 0) などのエンコーダが、マルチチャンネルオーディオのウィンドウを、後続のエンコーディングのためにタイルにグループ化する。これによって、フレームのチャンネルのさまざまな組合せに対するマルチチャンネル変換を可能にしながら、エンコーダに、フレームの異なるチャンネルについて異なるウィンドウ構成を使用する柔軟性が与えられる。図 7 のデコーダ (7 0 0) などのデコーダが、デコード中にタイルを処理する。

【 0 1 4 0 】

各チャンネルが、他のチャンネルと独立のウィンドウ構成を有することができる。同一の開始時刻および停止時刻を有するウィンドウは、タイルの一部とみなされる。タイルは、1 つまたは複数のチャンネルを有することができ、エンコーダは、タイル内のチャンネルに関してマルチチャンネル変換を実行する。

【 0 1 4 1 】

図 1 1 a に、ステレオオーディオのフレームの例のタイル構成 (1 1 0 0) を示す。図 1 1 a では、各タイルに単一のウィンドウが含まれる。ステレオオーディオのどちらのチャンネルのウィンドウも、他のチャンネルのウィンドウと同一の時刻に始まらず、停止しない。

【 0 1 4 2 】

図11bに、5.1チャンネルオーディオのフレームの例のタイル構成(1101)を示す。タイル構成(1101)には、0から6までの番号をつけられた7つのタイルが含まれる。タイル0には、チャンネル0、2、3、および4からのサンプルが含まれ、タイル0は、フレームの最初の1/4にまたがる。タイル1には、チャンネル1からのサンプルが含まれ、タイル1は、フレームの最初の1/2にまたがる。タイル2には、チャンネル5からのサンプルが含まれ、タイル2は、フレーム全体にまたがる。タイル3は、タイル0に似ているが、フレームの第2の1/4にまたがる。タイル4および6には、チャンネル0、2、および3のサンプルが含まれ、タイル4および6は、それぞれ、フレームの3番目の1/4および4番目の1/4にまたがる。最後に、タイル5には、チャンネル1および4からのサンプルが含まれ、タイル5は、フレームの後半分にまたがる。図11bからわかるように、特定のタイルに、不連続なチャンネルのウィンドウを含めることができる。

10

【0143】

図12に、マルチチャンネルオーディオのフレームのタイルを構成する一般化された手法(1200)を示す。エンコーダは、フレーム内のチャンネルに関してウィンドウ構成をセットし(1210)、各チャンネルを可変サイズウィンドウに区分して、時間分解能と周波数分解能をトレードオフする。たとえば、エンコーダのパーティショナ/タイルコンフィギュアラが、フレーム内の他のチャンネルと独立に各チャンネルを区分する。

【0144】

次に、エンコーダは、異なるチャンネルからのウィンドウをフレームのタイルにグループ化する(1220)。たとえば、エンコーダは、ウィンドウが同一の開始位置および同一の終了位置を有する場合に、異なるチャンネルからのウィンドウを単一のタイルに置く。代替案では、エンコーダは、異なるチャンネルのどの部分を一緒にタイルにグループ化するかを判定するのに、開始位置/終了位置以外の判断基準を使用するか、開始位置/終了位置に加えて判断基準を使用することができる。

20

【0145】

一実施形態では、エンコーダが、フレームに関するウィンドウ構成のセット(1210)の後に(それと独立に)、タイルのグループ化(1220)を実行する。他の実施形態では、エンコーダが、ウィンドウ構成をセットする(1210)のと同時にウィンドウをタイルにグループ化し(1220)て、たとえば、時間相関を優先する(より長いウィンドウを使用するか)、チャンネル相関を優先する(より多くのチャンネルを単一のタイルに置く)か、強制的にウィンドウを特定のタイルの組にあてはめることによってタイルの個数を制御する。

30

【0146】

次に、エンコーダは、エンコードされたオーディオデータと共に出力するために、フレームのタイル構成情報を送る(1230)。たとえば、エンコーダのパーティショナ/タイルコンフィギュアラが、タイルサイズおよびタイルのチャンネルメンバ情報をMUXに送る。代替案では、エンコーダが、タイル構成を指定する他の情報を送る。一実施形態では、エンコーダが、タイルグループ化(1220)の後にタイル構成情報を送る(1230)。他の実施形態では、エンコーダが、これらの処置を同時に実行する

図13に、特定のビットストリーム構文による、マルチチャンネルオーディオのフレームに関してタイルを構成し、タイル構成情報を送る手法(1300)を示す流れ図である。図13には、情報をビットストリームに入れるためにエンコーダによって実行される手法(1300)が示され、デコーダは、対応する手法(フラグを読み取る、特定のタイルに関する構成情報を得る、など)を実行して、ビットストリーム構文に従ってフレームのタイル構成情報を検索する。代替案では、デコーダおよびエンコーダが、図13に示されたオプションの1つまたは複数に関する別の構文、たとえば、異なるフラグまたは異なる順序付けを使用する構文を使用する。

40

【0147】

エンコーダは、当初は、フレームのチャンネルのどれもがウィンドウに分割されないかどうかを検査する(1310)。そうである場合には、エンコーダは、フラグビット(どの

50

チャンネルも分割されないことを示す)を送り(1312)、終了する。したがって、単一のビットによって、所与のフレームが単一のタイルであるか複数のタイルを有するかが示される。

【0148】

その一方で、少なくとも1つのチャンネルがウィンドウに分割される場合に、エンコーダは、フレームのすべてのチャンネルが同一のウィンドウ構成を有するか否かを検査する(1320)。そうである場合には、エンコーダは、フラグビット(すべてのチャンネルが同一のウィンドウ構成を有し、フレームの各タイルがすべてのチャンネルを有することを示す)とタイルサイズのシーケンスとを送り(1322)、終了する。したがって、単一のビットによって、チャンネルのすべてが同一の構成を有する(通常のエンコーダビットストリームと同様に)か、柔軟なタイル構成を有するかが示される。

10

【0149】

少なくともいくつかのチャンネルが異なるウィンドウ構成を有する場合に、エンコーダは、フレームのサンプル位置をスキャンして、同一の開始位置および同一の終了位置の両方を有するウィンドウを識別する。しかし、まず、エンコーダは、フレームのすべてのサンプル位置をグループ化されないものとしてマークする(1330)。次に、エンコーダは、チャンネル/時間スキャンパターンに従って、フレームの次のグループ化されていないサンプル位置をスキャンする(1340)。一実施形態では、エンコーダが、グループ化されていないサンプル位置を探して特定の時刻のすべてのチャンネルをスキャンし、その後、時間的に次のサンプル位置について繰り返す。他の実施形態では、エンコーダが、別のスキャンパターンを使用する。

20

【0150】

検出されたグループ化されていないサンプル位置について、エンコーダは、類似するウィンドウと一緒にタイルにグループ化する(1350)。具体的に言うと、エンコーダは、検出されたグループ化されていないサンプル位置を含むウィンドウの開始位置で始まり、検出されたグループ化されていないサンプル位置を含むウィンドウと同一の位置で終わるウィンドウをグループ化する。たとえば、図11bに示されたフレームでは、エンコーダは、まず、チャンネル0の先頭でサンプル位置を検出する。エンコーダは、チャンネル0、2、3、および4からの1/4フレーム長のウィンドウを、一緒にタイルにグループ化する。というのは、これらのウィンドウのそれぞれが、タイルの他のウィンドウと同一の開

30

【0151】

次に、エンコーダは、エンコードされたオーディオデータと共に出力するために、タイルを指定するタイル構成情報を送る(1360)。タイル構成情報には、タイルサイズと、タイル内のその点でグループ化されていないサンプル位置を有するどのチャンネルがタイルに含まれるかを示すマップが含まれる。チャンネルマップには、タイルに可能なチャンネルごとに1ビットを含めることができる。タイル情報のシーケンスに基づいて、デコーダは、タイルがフレーム内で始まり、終わるかどうかを判定する。エンコーダは、どのチャンネルがタイルに存在することができるかを考慮に入れることによって、チャンネルのビットレートを下げる。たとえば、図11bのタイル0の情報には、タイルサイズと、チャンネル0、2、3、および4がタイルの一部であることを示すバイナリパターン「101110」が含まれる。その点の後で、チャンネル1および5のサンプル位置だけが、グループ化されていない。したがって、タイル1の情報には、タイルサイズと、チャンネル1がタイルの一部であるが、チャンネル5がそうでないことを示すバイナリパターン「10」が含まれる。これによって、バイナリパターンの4ビットが節約される。次に、タイル2のタイル情報に、タイルサイズだけが含まれる(チャンネルマップは含まれない)。というのは、チャンネル5が、タイル2で始まるウィンドウを有することができる唯一のチャンネルであるからである。タイル3のタイル情報には、タイルサイズと、バイナリパターン「1111」が含まれる。というのは、チャンネル1および5が、タイル3の範囲内のグループ化された位置を有するからである。代替案では、エンコーダおよびデコーダが、別の手法を使用して、

40

50

構文でチャンネルパターンを知らせる。

【 0 1 5 2 】

次に、エンコーダは、タイルに含まれるウィンドウのサンプル位置を、グループ化されたものとしてマークし(1 3 7 0)、継続するか否かを判定する(1 3 8 0)。グループ化されていないサンプル位置がフレームにない場合には、エンコーダは終了する。そうでない場合には、エンコーダは、チャンネル/時間スキャンパターンに従って、フレームの次のグループ化されていないサンプル位置をスキャンする(1 3 4 0)。

【 0 1 5 3 】

V . 柔軟なマルチチャンネル変換

いくつかの実施形態で、図 6 のエンコーダ(6 0 0)などのエンコーダが、相互チャンネル相関性を効果的に活用する柔軟なマルチチャンネル変換を実行する。図 7 のデコーダ(7 0 0)などのデコーダが、対応する逆マルチチャンネル変換を実行する。

【 0 1 5 4 】

具体的に言うと、エンコーダおよびデコーダは、下記の 1 つまたは複数を行って、異なる状況でマルチチャンネル変換を改善する。

【 0 1 5 5 】

1 . エンコーダは、知覚的重みづけの後にマルチチャンネル変換を実行し、デコーダは、逆重みづけの前に、対応する逆マルチチャンネル変換を実行する。これによって、逆マルチチャンネル変換後のチャンネルにまたがる量子化ノイズのアンマスキングが減る。

【 0 1 5 6 】

2 . エンコーダおよびデコーダは、マルチチャンネル変換のためにチャンネルをグループ化して、どのチャンネルが一緒に変換されるかを制限する。

【 0 1 5 7 】

3 . エンコーダおよびデコーダは、どの帯域が一緒に変換されるかを制御するために、周波数帯域レベルでマルチチャンネル変換を選択的にオン/オフにする。

【 0 1 5 8 】

4 . エンコーダおよびデコーダは、階層マルチチャンネル変換を使用して、計算の複雑さを(特にデコーダで)制限する。

【 0 1 5 9 】

5 . エンコーダおよびデコーダは、事前定義のマルチチャンネル変換行列を使用して、変換行列の指定に使用されるビットレートを減らす。

【 0 1 6 0 】

6 . エンコーダおよびデコーダは、ビット効率のために、量子化されたギブンス回転ベースの因数分解パラメータを使用して、マルチチャンネル変換行列を指定する。

【 0 1 6 1 】

A . 重みづけされたマルチチャンネルオーディオに対するマルチチャンネル変換

いくつかの実施形態で、エンコーダは、知覚的重みづけの後にマルチチャンネル変換を位置付け(デコーダは、逆重みづけの前に逆マルチチャンネル変換を位置付け)、チャンネル間の漏れ信号が、制御され、測定可能であり、元の信号に類似するスペクトルを有するようにする。

【 0 1 6 2 】

図 1 4 に、エンコーダで知覚的重みづけの後に 1 つまたは複数のマルチチャンネル変換を実行する手法(1 4 0 0)を示す。エンコーダは、マルチチャンネルオーディオに知覚的に重みをつけ(1 4 1 0)、たとえば、周波数領域のマルチチャンネルオーディオに重みづけ係数を適用する。いくつかの実施形態で、エンコーダは、マルチチャンネル変換の前に、重みづけ係数とチャンネルごとの量子化ステップ変更子の両方をマルチチャンネルオーディオデータに適用する。

【 0 1 6 3 】

次に、エンコーダは、たとえば下で説明するように、重みをつけられたオーディオデータに対する 1 つまたは複数のマルチチャンネル変換を実行する(1 4 2 0)。最後に、エン

10

20

30

40

50

コーダは、マルチチャンネル変換されたオーディオデータを量子化する(1430)。

【0164】

図15に、デコーダで逆重みづけの前に逆マルチチャンネル変換を実行する手法(1500)を示す。デコーダは、たとえば下で説明するように、量子化されたオーディオデータに対して1つまたは複数の逆マルチチャンネル変換を実行する(1510)。具体的に言うと、デコーダは、特定の周波数インデックスの複数のチャンネルからのサンプルをベクトル x_{m_c} に集め、逆マルチチャンネル変換 A_{m_c} を実行して、出力 y_{m_c} を生成する。

$$y_{m_c} = A_{m_c} \cdot x_{m_c} \quad (7)$$

【0165】

その後、デコーダは、マルチチャンネルオーディオを逆量子化し、逆重みづけし(1520)、マスクによって逆マルチチャンネル変換の出力をカラーリングする。したがって、チャンネルにまたがって発生する(量子化に起因する)漏れが、スペクトルにおいて整形され、その結果、漏れた信号の可聴性が、測定可能かつ制御可能であり、所与の再構成されたチャンネルでの他のチャンネルの漏れが、所与のチャンネルの元の壊されない信号と同様にスペクトルにおいて整形される(いくつかの実施形態で、チャンネルごとの量子化ステップサイズ変更子によって、エンコーダが、再構成される信号の品質がすべての再構成されるチャンネルにまたがってほぼ同一になるようにすることを可能にすることもできる)。

【0166】

B. チャンネルグループ

いくつかの実施形態で、エンコーダおよびデコーダが、マルチチャンネル変換のためにチャンネルをグループ化して、一緒に変換されるチャンネルを制限する。たとえば、タイル構成を使用する実施形態では、エンコーダが、タイルのどのチャンネルが関連するかを判定し、関連するチャンネルをグループ化する。代替案では、エンコーダおよびデコーダが、タイル構成を使用しないが、フレームまたは他のレベルでチャンネルをグループ化する。

【0167】

図16に、一実施形態でマルチチャンネル変換についてタイルのチャンネルをグループ化する手法(1600)を示す。この手法(1600)では、エンコーダが、チャンネルの信号の間の対単位の相関ならびにいくつかの場合に帯域の間の相関を考慮する。代替案では、エンコーダが、マルチチャンネル変換についてチャンネルをグループ化する時に、他のおよび/または追加の要因を考慮する。

【0168】

まず、エンコーダは、タイルのチャンネルを得る(1610)。たとえば、図11bに示されたタイル構成では、タイル3が、その中に4つのチャンネルすなわち0、2、3、および4を有する。

【0169】

エンコーダは、チャンネルの信号の間の対単位の相関を計算し(1620)、それ相応にチャンネルをグループ化する(1630)。図11bのタイル3について、チャンネル0および2が、対単位で相関するが、この両方のチャンネルが、チャンネル3またはチャンネル4と対単位で相関せず、チャンネル3が、対単位でチャンネル4と相関しないと仮定する。エンコーダは、チャンネル0および2を一緒にグループ化し(1630)、チャンネル3を別のグループに入れ、チャンネル4をさらに別のグループに入れる。

【0170】

グループのどのチャンネルとも対単位で相関しないチャンネルが、それでもそのグループとの互換性を有する場合がある。したがって、グループとの互換性がないチャンネルについて、エンコーダは、任意選択として、帯域レベルでの互換性を検査し(1640)、それ相応にチャンネルの1つまたは複数のグループを調整する(1650)。具体的に言うと、これによって、ある帯域でグループとの互換性があるが、他の帯域で非互換であるチャンネルが識別される。たとえば、図11bのタイル3のチャンネル4が、実際にはほとんどの帯域でチャンネル0および2と互換であるが、少数の帯域での非互換性のゆえに、対単位の相関結果が歪曲されると仮定する。エンコーダは、グループを調整して(1650)、チャネ

10

20

30

40

50

ル0、2、および4を一緒にし、チャンネル3をそれ自体のグループに残す。エンコーダは、いくつかのチャンネルが「全体的に」関連するが、非互換帯域を有する時に、このようなテストを実行することもできる。これらの非互換帯域で変換をオフにすることによって、実際にマルチチャンネル変換コーディングされる帯域の間の相関が改善され、したがって、コーディング効率が改善される。

【0171】

所与のタイルのチャンネルは、1つのチャンネルグループに属する。チャンネルグループのチャンネルが、連続的である必要はない。単一のタイルに、複数のチャンネルグループを含めることができ、各チャンネルグループが、異なる関連するマルチチャンネル変換を有することができる。どのチャンネルが互換性を有するかを判断した後に、エンコーダは、チャンネルグループ情報をビットストリームに入れる。

10

【0172】

図17に、エンコーダがチャンネルグループを計算する方法に関係のない、特定のビットストリーム構文によるビットストリームからのタイルのチャンネルグループ情報およびマルチチャンネル変換情報の検索の手法(1700)を示す。図17には、ビットストリームから情報を検索するためにデコーダによって実行される手法(1700)が示され、エンコーダは、対応する手法を実行して、ビットストリーム構文に従って、タイルのチャンネルグループ情報およびマルチチャンネル変換情報をフォーマットする。代替案では、デコーダおよびエンコーダが、図17に示されたオプションの1つまたは複数について別の構文を使用する。

20

【0173】

まず、デコーダは、手法(1700)で使用される複数の変数を初期化する。デコーダは、タイル#ChannelsInTileのチャンネル数と等しくなるように#ChannelsToVisitをセットし(1710)、チャンネルグループ数#ChannelGroupsに0をセットする(1712)。

【0174】

デコーダは、#ChannelsToVisitが2を超えるかどうかを検査する(1720)。そうでない場合には、デコーダは、#ChannelsToVisitが2と等しいかどうかを検査する(1730)。そうである場合には、デコーダは、たとえば下で説明する手法を使用して、2チャンネルのグループのマルチチャンネル変換をデコードする(1740)。構文では、各チャンネルグループが、異なるマルチチャンネル変換を有することができる。その一方で、#ChannelsToVisitが1または0と等しい場合には、デコーダは、マルチチャンネル変換をデコードせずに終了する。

30

【0175】

#ChannelsToVisitが2を超える場合には、デコーダは、タイルのグループのチャンネルマスクをデコードする(1750)。具体的には、デコーダは、チャンネルマスクのビットストリームから#ChannelsToVisitビットを読み取る。チャンネルマスクの各ビットによって、特定のチャンネルがチャンネルグループに含まれるか否かが示される。たとえば、チャンネルマスクが「10110」である場合に、タイルに5つのチャンネルが含まれ、チャンネル0、2、および3がチャンネルグループに含まれる。

40

【0176】

デコーダは、グループのチャンネル数をカウントし(1760)、たとえば下で説明する手法を使用して、グループのマルチチャンネル変換をデコードする(1770)。デコーダは、現在のチャンネルグループのカウントされたチャンネル数を引くことによって#ChannelsToVisitを更新し(1780)、#ChannelGroupsを増分し(1790)、視察すべき残されたチャンネル数#ChannelsToVisitが2を超えるかどうかを検査する(1720)。

【0177】

代替案では、タイル構成を使用しない実施形態で、デコーダが、フレームまたは他のレベルに関するチャンネルグループ情報およびマルチチャンネル変換情報を検索する。

50

【 0 1 7 8 】

C . マルチチャンネル変換の帯域オン / オフ制御

いくつかの実施形態で、エンコーダおよびデコーダが、周波数帯域レベルでマルチチャンネル変換を選択的にオン / オフにして、どの帯域と一緒に変換されるかを制御する。この形で、エンコーダおよびデコーダが、マルチチャンネル変換で互換性がない帯域を選択的に除外する。マルチチャンネル変換が、特定の帯域についてオフにされる時に、エンコーダおよびデコーダは、その帯域に恒等変換を使用し、データを変更せずにその帯域のデータを通過させる。

【 0 1 7 9 】

周波数帯域は、臨界帯域または量子化帯域である。周波数帯域の数は、オーディオデータのサンプリング周波数およびタイルサイズに関係する。一般に、サンプリング周波数が高くなるかタイルサイズが大きくなると、周波数帯域の数が増える。

10

【 0 1 8 0 】

いくつかの実施形態で、エンコーダが、タイルのチャンネルグループのチャンネルについて、周波数帯域レベルでマルチチャンネル変換を選択的にオン / オフにする。エンコーダは、タイルのチャンネルをグループ化する時またはタイルに関するチャンネルグループ化の後に、帯域をオン / オフにすることができる。代替案では、エンコーダおよびデコーダが、タイル構成を使用するのではなく、フレームまたは他のレベルについて周波数帯域でマルチチャンネル変換をオン / オフにする。

【 0 1 8 1 】

20

図 1 8 に、一実施形態でマルチチャンネル変換にチャンネルグループのチャンネルの周波数帯域を選択的に含める手法 (1 8 0 0) を示す。手法 (1 8 0 0) では、エンコーダが、帯域のチャンネルの信号の間の対単位の相関を検討して、その帯域のマルチチャンネル変換を使用可能にするか使用不能にするかを判定する。代替案では、エンコーダが、マルチチャンネル変換について周波数帯域を選択的にオンまたはオフにする時に、他のおよび / または追加の要因を検討する。

【 0 1 8 2 】

まず、エンコーダは、たとえば図 1 6 に関して説明したように、チャンネルグループのチャンネルを入手する (1 8 1 0) 。次に、エンコーダは、異なる周波数帯域のチャンネルの信号の間の対単位の相関を計算する (1 8 2 0) 。たとえば、チャンネルグループに 2 つのチャンネルが含まれる場合に、エンコーダは、各周波数帯域での対単位の相関を計算する。あるいは、チャンネルグループに 2 つを超えるチャンネルが含まれる場合に、エンコーダは、各周波数帯域でのめいめいのチャンネル対の一部またはすべての間の対単位の相関を計算する。

30

【 0 1 8 3 】

次に、エンコーダは、チャンネルグループのマルチチャンネル変換について、帯域をオンまたはオフにする (1 8 3 0) 。たとえば、チャンネルグループに 2 つのチャンネルが含まれる場合に、エンコーダは、帯域での対単位の相関が特定の閾値を満足する場合に、その帯域のマルチチャンネル変換を使用可能にする。あるいは、チャンネルグループに 2 つを超えるチャンネルが含まれる場合に、エンコーダは、帯域の対単位の相関のそれぞれまたは大多数が特定の閾値を満足する場合に、その帯域のマルチチャンネル変換を使用可能にする。代替実施形態では、すべてのチャンネルについて特定の周波数帯域をオンまたはオフにするのではなく、エンコーダが、帯域を、あるチャンネルについてオン、他のチャンネルについてオフにする。

40

【 0 1 8 4 】

どの帯域がマルチチャンネル変換に含まれるかを判断した後に、エンコーダは、帯域オン / オフ情報をビットストリームに入れる。

【 0 1 8 5 】

図 1 9 に、エンコーダがどのように帯域をオンまたはオフにすると判断するかに無関係に、特定のビットストリーム構文によるビットストリームからのタイルのチャンネルグルー

50

プに関するマルチチャンネル変換の帯域オン/オフ情報を検索する手法(1900)を示す。図19には、ビットストリームから情報を検索するためにデコーダによって実行される手法(1900)が示され、エンコーダは、対応する手法を実行して、ビットストリーム構文に従ってチャンネルグループの帯域オン/オフ情報をフォーマットする。代替案では、デコーダおよびエンコーダが、図19に示されたオプションの1つまたは複数について別の構文を使用する。

【0186】

いくつかの実施形態で、デコーダは、手法(1700)のマルチチャンネル変換のデコード(1740または1770)の一部として手法(1900)を実行する。代替案では、デコーダが、手法(1900)を別々に実行する。

10

【0187】

デコーダは、ビットを入手し(1910)、ビットを検査して(1920)、チャンネルグループについて、すべての帯域が使用可能にされているかどうかを判定する。そうである場合には、デコーダは、チャンネルグループのすべての帯域についてマルチチャンネル変換を使用可能にする(1930)。

【0188】

その一方で、ビットによって、チャンネルグループのすべての帯域が使用可能にされているのではないことが示される場合に、デコーダは、チャンネルグループの帯域マスクをデコードする(1940)。具体的に言うと、デコーダは、ビットストリームからビット数を読み取るが、この数は、チャンネルグループの帯域の数である。帯域マスクの各ビットが、特定の帯域がチャンネルグループについてオンまたはオフのどちらであることを示す。たとえば、帯域マスクが「111111110110000」である場合には、チャンネルグループに15個の帯域が含まれ、帯域0、1、2、3、4、5、6、7、9、および10が、マルチチャンネル変換についてオンにされている。デコーダは、示された帯域についてマルチチャンネル変換を使用可能にする(1950)。

20

【0189】

その代わりに、タイル構成を使用しない実施形態では、デコーダが、フレームまたは他のレベルでの帯域オン/オフ情報を検索する。

【0190】

D. 階層マルチチャンネル変換

30

いくつかの実施形態で、エンコーダおよびデコーダは、階層マルチチャンネル変換を使用して、特にデコーダでの、計算の複雑さを制限する。階層変換を用いるときに、エンコーダが、全体的な変換を複数のステージに分割し、個々のステージの計算の複雑さを減らし、いくつかの場合にマルチチャンネル変換を指定するのに必要な情報の量を減らす。このカスケード構造を使用して、エンコーダは、より大きい全体的な変換を、ある精度まで、より小さい変換を用いてエミュレートする。デコーダは、対応する階層逆変換を実行する。

【0191】

いくつかの実施形態で、階層変換の各ステージが、構造において同一であり、ビットストリーム内で、各ステージが、1つまたは複数の他のステージと独立に記述される。具体的に言うと、各ステージが、それ自体のチャンネルグループと、チャンネルグループごとに1つのマルチチャンネル変換行列を有する。代替実施形態では、異なるステージが、異なる構造を有し、エンコーダおよびデコーダで、異なるビットストリーム構文が使用され、かつ/またはステージで、チャンネルおよび変換に関する別の構成が使用される。

40

【0192】

図20に、より単純なマルチチャンネル変換の階層を使用してマルチチャンネル変換をエミュレートする一般化された手法(2000)を示す。図20には、nステージの階層が示され、nは、マルチチャンネル変換ステージの数である。たとえば、一実施形態で、nは2である。代替案では、nが2より大きい。

【0193】

エンコーダは、全体的な変換のマルチチャンネル変換の階層を判定する(2010)。エ

50

ンコーダは、逆変換を実行するデコーダの複雑さに基づいて、変換サイズ（すなわち、チャンネルグループサイズ）を判断する。あるいは、エンコーダは、ターゲットデコーダプロファイル/デコーダレベルまたは他の判断基準を検討する。

【 0 1 9 4 】

図 2 1 は、マルチチャンネル変換の例の階層（ 2 1 0 0 ）を示す図である。この階層（ 2 1 0 0 ）には、2つのステージが含まれる。第 1 ステージには、0 から N までの番号をつけられた、N + 1 個のチャンネルグループおよび変換が含まれ、第 2 ステージには、0 から M までの番号をつけられた、M + 1 個のチャンネルグループおよび変換が含まれる。各チャンネルグループに、1 つまたは複数のチャンネルが含まれる。第 1 ステージの N + 1 個の変換のそれぞれについて、入力チャンネルは、マルチチャンネルトランスフォーマーに入力されるチャンネルのある組合せである。すべての入力チャンネルを第 1 ステージで変換しなければならないわけではない。1 つまたは複数の入力チャンネルを、無変更で第 1 ステージを通過させることができる（たとえば、エンコーダによって、チャンネルグループに含まれる、単位行列を使用するチャンネルを含めることができる）。第 2 ステージの M + 1 個の変換のそれぞれについて、入力チャンネルは、第 1 ステージからの出力チャンネルのある組合せであり、この出力チャンネルには、無変更で第 1 ステージを通過した可能性があるチャンネルが含まれる。

10

【 0 1 9 5 】

図 2 0 に戻って、エンコーダは、マルチチャンネル変換の第 1 ステージを実行し（ 2 0 2 0 ）、マルチステージ変換の次のステージを実行し、最後に、マルチチャンネル変換の第 n ステージを実行する（ 2 0 3 0 ）。デコーダは、デコード中に、対応する逆マルチチャンネル変換を実行する。

20

【 0 1 9 6 】

いくつかの実施形態で、チャンネルグループが、階層の複数のステージで同一であるが、マルチチャンネル変換は異なる。そのような場合、およびいくつかの他の場合に、エンコーダは、複数のマルチチャンネル変換について周波数帯域オン/オフ情報を組み合わせることができる。たとえば、2つのマルチチャンネル変換があり、それぞれのチャンネルグループに同一の3つのチャンネルがあると仮定する。エンコーダは、帯域 0 の両方のステージで変換なし/恒等変換、帯域 1 のマルチチャンネル変換ステージ 1 のみ（ステージ 2 変換なし）、帯域 2 のマルチチャンネル変換ステージ 2 のみ（ステージ 1 変換なし）、帯域 3 の両方のステージのマルチチャンネル変換、帯域 4 の両方のステージでの変換なしなどを指定することができる。

30

【 0 1 9 7 】

図 2 2 に、特定のビットストリーム構文によるビットストリームからのチャンネルグループに関するマルチチャンネル変換の階層の情報を検索する手法（ 2 2 0 0 ）を示す。図 2 2 には、ビットストリームを解析するためにデコーダによって実行される手法（ 2 2 0 0 ）が示され、エンコーダは、対応する手法を実行して、ビットストリーム構文に従ってマルチチャンネル変換の階層をフォーマットする。代替案では、デコーダおよびエンコーダが、別の構文、たとえば、2つを超えるステージのための追加フラグおよびシグナリングビットを含む構文を使用する。

40

【 0 1 9 8 】

デコーダは、まず、ビットストリームの次のビットと等しくなるように一時値 i_{Tmp} をセットする（ 2 2 1 0 ）。次に、デコーダは、一時値の値を検査し（ 2 2 2 0 ）、この値によって、デコーダが、ステージ 1 グループのチャンネルグループおよびマルチチャンネル変換情報をデコード（ 2 2 3 0 ）しなければならないか否かが知らされる。

【 0 1 9 9 】

デコーダは、ステージ 1 グループのチャンネルグループおよびマルチチャンネル変換情報をデコード（ 2 2 3 0 ）した後に、ビットストリームの次のビットと等しくなるように i_{Tmp} をセットする（ 2 2 4 0 ）。デコーダは、 i_{Tmp} の値を検査する（ 2 2 2 0 ）が、この値によって、さらなるステージ 1 グループに関するチャンネルグループおよびマルチチ

50

チャンネル変換情報がビットストリームに含まれるか否かが知らされる。恒等変換を有しないチャンネルグループだけが、ビットストリームのステージ1部分で指定され、ビットストリームのステージ1部分に記載されていないチャンネルは、恒等変換を使用するチャンネルグループの一部と仮定される。

【0200】

ビットストリームに、ステージ1グループのチャンネルグループおよびマルチチャンネル変換情報がこれ以上含まれない場合には、デコーダは、すべてのステージ2グループのチャンネルグループおよびマルチチャンネル変換情報をデコードする(2250)。

【0201】

E. 事前定義またはカスタムのマルチチャンネル変換

いくつかの実施形態で、エンコーダおよびデコーダが、事前定義のマルチチャンネル変換行列を使用して、変換行列の指定に使用されるビットレートを減らす。エンコーダは、複数の使用可能な事前定義の行列タイプの中から選択し、選択された行列を、ビットストリーム内で少数(たとえば、1、2)のビットを使用して知らせる。行列のタイプの中には、ビットストリーム内の追加シグナリングを必要としないものと、追加の指定を必要とするものがある。デコーダは、行列のタイプを示す情報と(必要な場合に)行列を指定する追加情報を検索する。

【0202】

いくつかの実施形態で、エンコーダおよびデコーダが、下記の事前定義行列タイプを使用する: 恒等、アダマール、DCTタイプII、または任意のユニタリ。代替案では、エンコーダおよびデコーダが、異なるおよび/または追加の事前定義行列タイプを使用する。

【0203】

図9aに、別のコンテキストでの6つのチャンネルの単位行列の例が示されている。単位行列の次元の数が、他の情報(たとえば、グループのチャンネル数)からエンコーダおよびデコーダに既知になると仮定して、エンコーダは、フラグビットを使用して、ビットストリームで単位行列を効率的に指定する。

【0204】

アダマール行列は、下記の形を有する。

【0205】

【数6】

$$A_{\text{Hadamard}} = \rho \begin{bmatrix} 0.5 & -0.5 \\ 0.5 & 0.5 \end{bmatrix} \quad (8)$$

【0206】

ここで、 ρ は、正規化スケーラ

【0207】

【数7】

$$(\sqrt{2})$$

【0208】

である。エンコーダは、ステレオデータのアダマール行列を、ビットストリーム内でフラグビットを使用して効率的に指定する。

【0209】

DCTタイプII行列は、下記の形を有する。

【0210】

10

20

30

40

【数 8】

$$A_{\text{DCT,II}} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{N-1,0} & a_{N-1,1} & \cdots & a_{N-1,N-1} \end{bmatrix} \quad (9)$$

【 0 2 1 1 】

ここで

【 0 2 1 2 】

10

【数 9】

$$a_{n,m} = k_m \cdot \cos\left(\frac{m \cdot (n + 0.5)\pi}{N}\right) \quad (10)$$

【 0 2 1 3 】

また、

【 0 2 1 4 】

【数 1 0】

$$k_m = \begin{cases} \sqrt{\frac{1}{N}} & m = 0 \\ \sqrt{\frac{2}{N}} & m > 0 \end{cases} \quad (11) \quad 20$$

【 0 2 1 5 】

である。

【 0 2 1 6 】

DCTタイプII行列に関する追加情報については、文献を参照されたい(たとえば、非特許文献4参照)。DCTタイプII行列は、任意のサイズを有することができる(すなわち、すべてのサイズのチャンネルグループについて働く)。DCTタイプII行列の次元の数が、他の情報(たとえば、グループのチャンネル数)からエンコーダおよびデコーダに既知になると仮定して、エンコーダは、フラグビットを使用して、ビットストリームで単位行列を効率的に指定する。

30

正方行列 A_{square} は、その転置行列が逆行列である場合に、ユニタリである。

$$A_{\text{square}} \cdot A_{\text{square}}^T = A_{\text{square}}^T \cdot A_{\text{square}} = I \quad (12)$$

ここで、 I は、単位行列である。エンコーダは、任意のユニタリ行列を使用して、効果的な冗長性除去のためのKLT変換を指定する。エンコーダは、ビットストリーム内で、フラグビットおよび行列のパラメータ化を使用して、任意のユニタリ行列を効率的に指定する。いくつかの実施形態で、エンコーダは、下で説明するように、量子化されたギブンス因数分解回転を使用して行列をパラメータ化する。代替案では、エンコーダが、別のパラメータ化を使用する。

40

【 0 2 1 7 】

図23に、複数の使用可能なタイプの中からマルチチャンネル変換タイプを選択する手法(2300)を示す。エンコーダは、チャンネルグループごとにまたはある他のレベルで、変換タイプを選択する。

【 0 2 1 8 】

エンコーダは、複数の使用可能なタイプの中からマルチチャンネル変換タイプを選択する(2310)。たとえば、使用可能なタイプに、恒等、アダマール、DCTタイプII、および任意のユニタリが含まれる。代替案では、タイプに、異なるおよび/または追加の行列タイプが含まれる。エンコーダは、可能な場合、または変換行列を指定するのに必要

50

なビット数を減らすのに必要な場合に、恒等行列、アダマール行列、またはDCTタイプII行列（任意のユニタリ行列ではなく）を使用する。たとえば、エンコーダは、冗長性除去が、任意のユニタリ行列による冗長性除去に匹敵するか十分に近い（ある判断基準によって）場合に、恒等行列、アダマール行列、またはDCTタイプII行列を使用する。あるいは、エンコーダは、ビットレートを削減しなければならない場合に、恒等行列、アダマール行列、またはDCTタイプII行列を使用する。しかし、一般的な状況で、エンコーダは、最良の圧縮効率のために任意のユニタリ行列を使用する。

【0219】

エンコーダは、選択されたタイプのマルチチャンネル変換を、マルチチャンネルオーディオデータに適用する（2320）。

10

【0220】

図24に、複数の使用可能なタイプの中からマルチチャンネル変換タイプを検索し、逆マルチチャンネル変換を実行する手法（2400）を示す。デコーダは、チャンネルグループごともまたは他のレベルで変換タイプ情報を検索する。

【0221】

デコーダは、複数の使用可能なタイプの間からマルチチャンネル変換タイプを検索する（2410）。たとえば、使用可能なタイプに、恒等、アダマール、DCTタイプII、および任意のユニタリが含まれる。代替案では、タイプに、異なるおよび/または追加の行列タイプが含まれる。必要な場合には、デコーダは、行列を指定する追加情報を検索する。

20

【0222】

行列を再構成した後に、デコーダは、選択されたタイプの逆マルチチャンネル変換をマルチチャンネルオーディオデータに適用する（2420）。

【0223】

図25に、特定のビットストリーム構文によるビットストリームからチャンネルグループに関するマルチチャンネル変換情報を検索する手法（2500）を示す。図25には、ビットストリームを解析するためにデコーダによって実行される手法（2500）が示されているが、エンコーダは、対応する手法を使用して、ビットストリーム構文に従ってマルチチャンネル変換情報をフォーマットする。代替案では、デコーダおよびエンコーダが、別の構文、たとえば、異なるフラグビット、異なる順序付け、または異なる変換タイプを使用する構文を使用する。

30

【0224】

当初、デコーダは、グループのチャンネル数#ChannelsInGroupが1より大きいかどうかを検査する（2510）。そうでない場合には、チャンネルグループがモノラルオーディオであり、デコーダは、グループに恒等変換を使用する（2512）。

【0225】

#ChannelsInGroupが1より大きい場合には、デコーダは、#ChannelsInGroupが2より大きいかどうかを検査する（2520）。そうでない場合には、チャンネルグループはステレオオーディオであり、デコーダは、ビットストリームの次のビットと等しくなるように一時値iTmpをセットする（2522）。次に、デコーダは、一時値の値を検査する（2524）が、この値によって、デコーダが、そのチャンネルグループにアダマール変換を使用（2530）しなければならないかが示される。そうでない場合には、デコーダは、ビットストリームの次のビットと等しくなるように一時値iTmpをセットし（2526）、iTmpの値を検査し（2528）、この値によって、デコーダが、チャンネルグループに恒等変換を使用（2550）しなければならないかが示される。そうでない場合には、デコーダは、チャンネルグループに汎用ユニタリ変換をデコードする（2570）。

40

【0226】

#ChannelsInGroupが2より大きい場合には、チャンネルグループは、サラウンドサウンドオーディオであり、デコーダは、ビットストリームの次のビットと等し

50

くなるように一時値 $i T m p$ をセットする (2 5 4 0) 。デコーダは、一時値の値を検査し (2 5 4 2) 、この値によって、デコーダが、チャンネルグループのサイズ $\# C h a n n e l s I n G r o u p$ の恒等変換を使用 (2 5 5 0) しなければならないかが示される。そうでない場合には、デコーダは、ビットストリームの次のビットと等しくなるように一時値 $i T m p$ をセットし (2 5 6 0) 、 $i T m p$ の値を検査する (2 5 6 2) 。このビットによって、デコーダが、チャンネルグループの汎用ユニタリ変換をデコード (2 5 7 0) しなければならないか、チャンネルグループのサイズ $\# C h a n n e l s I n G r o u p$ の D C T タイプ I I 変換を使用 (2 5 8 0) しなければならないかが示される。

【 0 2 2 7 】

デコーダは、チャンネルグループに関してアダマール変換行列、 D C T タイプ I I 変換行列、または汎用ユニタリ変換行列を使用する時に、行列のマルチチャンネル変換帯域オン/オフ情報をデコードし (2 5 9 0) 、終了する。

10

【 0 2 2 8 】

F . 変換行列のギブンス回転表現

いくつかの実施形態で、エンコーダおよびデコーダが、ビット効率のために、量子化されたギブンス回転ベースの因数分解パラメータを使用して、任意のユニタリ変換行列を指定する。

【 0 2 2 9 】

一般に、ユニタリ変換行列は、ギブンス因数分解回転を使用して表すことができる。この因数分解を使用すると、ユニタリ変換行列を、次のように表すことができる。

20

【 0 2 3 0 】

【 数 1 1 】

$$A_{\text{unitary}} = \Theta_{0,N-2} \cdots \Theta_{0,1} \Theta_{0,0} \Theta_{1,N-3} \cdots \Theta_{1,1} \Theta_{1,0} \cdots \Theta_{N-2,0} \begin{bmatrix} \alpha_0 & 0 & \cdots & 0 \\ 0 & \alpha_1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \alpha_{N-1} \end{bmatrix} \quad (13)$$

【 0 2 3 1 】

ここで、 θ_i は、 + 1 または - 1 (回転の符号) であり、各 θ_i は、図 2 6 に示された回転行列 (2 6 0 0) の形である。回転行列 (2 6 0 0) は、単位行列にほとんど似ているが、変化する位置に 4 つのサイン/コサイン項を有する。図 2 7 a から 2 7 c に、マルチチャンネル変換行列を表すギブンス回転の例の回転行列を示す。 2 つのコサイン項が、必ず対角線上にあり、 2 つのサイン項が、コサイン項と同一の行/列にある。各 θ_i は、 1 つの回転角度を有し、その値は、範囲

30

【 0 2 3 2 】

【 数 1 2 】

$$-\frac{\pi}{2} \leq \theta_k < \frac{\pi}{2}$$

40

【 0 2 3 3 】

を有することができる。 $N \times N$ ユニタリ行列 A_{unitary} を完全に記述するのに必要なそのような回転行列 の数は、次の通りである。

【 0 2 3 4 】

【 数 1 3 】

$$\frac{N(N-1)}{2} \quad (14)$$

【 0 2 3 5 】

ギブンス因数分解回転に関する追加情報については、参照によって本明細書に組み込ま

50

れる文献を参照されたい(たとえば、非特許文献5参照)。

【0236】

いくつかの実施形態で、エンコーダは、ギブンス因数分解の回転角度を量子化して、ビットレートを減らす。図28に、量子化されたギブンス因数分解回転を使用してマルチチャネル変換行列を表す手法(2800)を示す。代替案では、エンコーダまたは処理ツールが、量子化されたギブンス因数分解回転を使用して、オーディオチャネルのマルチチャネル変換以外の目的のユニタリ行列を表す。

【0237】

エンコーダは、まず、マルチチャネル変換の任意のユニタリ行列を計算する(2810)。

次に、エンコーダは、ユニタリ行列のギブンス因数分解回転を計算する(2820)。

10

【0238】

ビットレートを減らすために、エンコーダは、回転角度を量子化する(2830)。一実施形態では、エンコーダが、各回転角度を64個($2^6 = 64$)の可能な値の1つに均等に量子化する。回転の符号は、それぞれ1ビットによって表され、したがって、エンコーダは、下記の数のビットを使用して、 $N \times N$ ユニタリ行列を表す。

【0239】

【数14】

$$6 \cdot \frac{N(N-1)}{2} + N = 3N^2 - 2N \quad (15)$$

20

【0240】

このレベルの量子化を用いると、エンコーダが、非常によい度合の精度で、マルチチャネル変換の $N \times N$ ユニタリ行列を表せるようになる。代替案では、エンコーダが、ある他のレベルおよび/またはタイプの量子化を使用する。

【0241】

図29に、特定のビットストリーム構文によるビットストリームからチャネルグループの汎用ユニタリ変換の情報を検索する手法(2900)を示す。図29には、ビットストリームを解析するためにデコーダによって実行される手法(2900)が示され、エンコーダは、対応する手法を実行して、ビットストリーム構文に従って汎用ユニタリ変換の情報をフォーマットする。代替案では、デコーダおよびエンコーダが、別の構文、たとえば、異なる順序付けまたは回転角度の分解能を使用する構文を使用する。

30

【0242】

まず、デコーダは、デコードの残りで使用される複数の変数を初期化する。具体的に言うと、デコーダは、デコードする角度の数#AnglesToDecodeを、式14に示されたチャネルグループのチャネルの数#ChannelsInGroupに基づいてセットする(2910)。デコーダは、#ChannelsInGroupに基づいて、デコードする符号の数#SignsToDecodeもセットする(2912)。デコーダは、デコードされた角度のカウントiAnglesDecodedおよびデコードされた符号のカウントiSignsDecodedもリセットする(2914、2916)。

40

【0243】

デコーダは、デコードする角度があるかどうかを検査し(2920)、そうである場合には、次の回転角度の値をセットし(2922)、6ビットの量子化された値から回転角度を再構成する。

$$\text{RotationAngle}[i\text{AnglesDecoded}] = \text{getBits}(6) - 32) / 64 \quad (16)$$

【0244】

次に、デコーダは、デコードされた角度のカウントを増分し(2924)、さらにデコードする追加の角度があるかどうかを検査する(2920)。

【0245】

50

デコードする角度がもうない時に、デコーダは、デコードする追加の符号があるかどうかを検査し(2940)、そうである場合には、次の符号の値をセットし(2942)、1ビットの値から符号を再構成する。

$RotationSign[iSignsDecoded] = (2 \times getBits(1)) - 1$ (17)

【0246】

次に、デコーダは、デコードされた符号のカウンタを増分し(2944)、デコードする追加の符号があるかどうかを検査する(2940)。デコードする符号がもうない時に、デコーダは終了する。

【0247】

VI. 量子化および重みづけ

いくつかの実施形態で、図6のエンコーダ(600)などのエンコーダが、下で説明するさまざまな手法を使用して、オーディオデータに対する量子化および重みづけを実行する。タイルに構成されたマルチチャンネルオーディオに関して、エンコーダは、タイルのチャンネルの量子化行列、チャンネルごとの量子化ステップ変更子、および全体的な量子化タイル係数を計算し、適用する。これによって、エンコーダが、聴覚モデルに従って雑音を整形し、チャンネルの間の雑音のバランスをとり、全体的なひずみを制御できるようになる。

【0248】

図7のデコーダ(700)などの対応するデコーダは、逆量子化および逆重みづけを実行する。タイルに構成されたマルチチャンネルオーディオについて、デコーダは、全体的な量子化タイル係数、チャンネルごとの量子化ステップ変更子、およびタイルのチャンネルの量子化行列をデコードし、適用する。逆量子化および逆重みづけが、単一のステップに融合される。

【0249】

A. 全体的なタイル量子化係数

いくつかの実施形態で、タイルのオーディオデータの品質および/またはビットレートを制御するために、エンコーダのクオンタイザが、タイルの量子化ステップサイズ Q_t を計算する。クオンタイザは、レート/品質コントローラと共に働いて、ビットレートおよび/または品質制約を満足するタイル量子化ステップサイズを選択する前に、タイルの異なる量子化ステップサイズを評価することができる。たとえば、クオンタイザおよびコントローラは、参照によって本明細書に組み込まれる関連特許出願の発明の名称"Quality and Rate Control Strategy for Digital Audio,"の米国特許出願第10/017,694号(2001年12月14日出願)に記載されているように動作する。

【0250】

図30に、特定のビットストリーム構文によるビットストリームから全体的なタイル量子化係数を検索する手法(3000)を示す。図30には、ビットストリームを解析するためにデコーダによって実行される手法(300)が示され、エンコーダは、対応する手法を実行して、ビットストリーム構文に従ってタイル量子化係数をフォーマットする。代替案では、デコーダおよびエンコーダが、別の構文、たとえば、タイル量子化係数の異なる範囲を扱うもの、異なる論理を使用してタイル係数をエンコードするもの、またはタイル係数のグループをエンコードするものを使用する。

【0251】

まず、デコーダは、タイルの量子化ステップサイズ Q_t を初期化する(3010)。一実施形態では、デコーダは、 Q_t に下記をセットする。

$Q_t = 90 \cdot ValidBitsPerSample / 16$ (18)

ここで、ValidBitsPerSampleは、16 ValidBitsPerSample 24の数であり、デコーダまたはオーディオクリップについてセットされるか、他のレベルでセットされる。

【0252】

次に、デコーダは、 Q_t の初期値に関する Q_t の最初の修正を示す6ビットを入手し(

10

20

30

40

50

3020)、値 - 32 Tmp 31を一時変数Tmpに保管する。関数SignExtend()は、符号なしの値から符号付きの値を判定する。デコーダは、Tmpの値を Q_t の初期値に加算し(3030)、その後、変数Tmpの符号を判定し(3040)、この符号は、変数SignofDeltaに保管される。

【0253】

デコーダは、Tmpの値が - 32または31と等しいかどうかを検査する(3050)。そうでない場合には、デコーダは終了する。Tmpの値が - 32または31と等しい場合には、エンコーダは、 Q_t をさらに修正しなければならないことを知らされている。さらなる修正の方向(正または負)は、SignofDeltaによって示され、デコーダは、次の5ビットを得て(3060)、次の修正の大きさ0 Tmp 31を判定する。デコーダは、 Q_t の現在の値を、SignofDeltaの方向でTmpの値だけ変更し(3070)、Tmpの値が31であるかどうかを検査する(3080)。そうでない場合には、デコーダは終了する。Tmpの値が31である場合には、デコーダは、次の5ビットを得て(3060)、その点から継続する。

10

【0254】

タイル構成を使用しない実施形態では、エンコーダが、フレームまたはオーディオデータの他の部分に関する全体的な量子化ステップサイズを計算する。

【0255】

B. チャンネルごとの量子化ステップ変更子

いくつかの実施形態で、エンコーダは、タイルの各チャンネルの量子化ステップ変更子： $Q_{c,0}$ 、 $Q_{c,1}$ 、 \dots 、 $Q_{c,\#channelsintile-1}$ を計算する。エンコーダは、通常は、これらのチャンネル固有量子化係数を計算して、すべてのチャンネルにまたがる再構成品質のバランスをとる。タイル構成を使用しない実施形態であっても、エンコーダは、フレームまたはオーディオデータの他の単位でチャンネルのチャンネルごとの量子化係数を計算することができる。対照的に、図1のエンコーダ(100)で使用されるものなどの、以前の量子化手法は、チャンネル内のウィンドウの帯域ごとに量子化行列要素を使用するが、チャンネルに関する全体的な変更子を有しない。

20

【0256】

図31に、マルチチャンネルオーディオデータのチャンネルごとの量子化ステップ変更子を計算する一般化された手法(3100)を示す。エンコーダは、複数の判断基準を使用して、量子化ステップ変更子を計算する。第1に、エンコーダは、再構成されるオーディオデータのすべてのチャンネルにまたがってほぼ等しい品質を探す。第2に、スピーカ位置が既知である場合に、エンコーダは、スピーカ構成に関する通常の使用での知覚に最も重要なスピーカを優先する。第3に、スピーカタイプが既知である場合に、エンコーダは、スピーカ構成でのよりよいスピーカを優先する。代替案では、エンコーダが、これらの判断基準以外のまたはこれらの判断基準に加えて判断基準を考慮する。

30

【0257】

エンコーダは、チャンネルの量子化ステップ変更子をセットする(3110)ことによって開始する。一実施形態では、エンコーダは、めいめいのチャンネルのエネルギーに基づいて変更子をセットする(3110)。たとえば、他のチャンネルより相対的により多くのエネルギー(すなわち大音量)を有するチャンネルについて、他のチャンネルの量子化ステップ変更子が、比較的に大きくされる。代替案では、エンコーダが、「オープンループ」推定処理で、他のまたは追加の判断基準に基づいて変更子をセットする(3110)。あるいは、エンコーダは、変更子に当初は等しい値をセットする(3110)ことができる(変更子の最終的な値に集束するのに「クローズドループ」評価に頼ってセットする)。

40

【0258】

エンコーダは、量子化ステップ変更子ならびに、他の量子化(重みづけを含む)要因がまだ適用されていない場合にはそのような他の要因を使用して、マルチチャンネルオーディオデータを量子化する(3120)。

【0259】

50

後続の再構成の後に、エンコーダは、NERまたは他の品質測定値を使用して、再構成されたオーディオのチャンネルの品質を評価する(3130)。エンコーダは、再構成されたオーディオが品質判断基準(および/または他の判断基準)を満足するか否かを検査し(3140)、そうである場合には終了する。そうでない場合には、エンコーダは、量子化ステップ変更子の新しい値をセットし(3110)、評価された結果に鑑みて変更子を調節する。代替案では、ステップ変更子の1パスのオープンループ設定について、エンコーダが、評価(3130)および検査(3140)をスキップする。

【0260】

チャンネルごとの量子化ステップ変更子は、ウィンドウ/タイルからウィンドウ/タイルへと変化する傾向を有する。エンコーダは、リテラルまたは可変長コードとして量子化ステップ変更子をコーディングし、それをオーディオデータと共にビットストリームにパックする。あるいは、エンコーダは、他の手法を使用して、量子化ステップ変更子を処理する。

10

【0261】

図32に、特定のビットストリーム構文によるビットストリームからチャンネルごとの量子化ステップ変更子を検索する手法(3200)を示す。図32には、ビットストリームを解析するためにデコーダによって実行される手法(3200)が示され、エンコーダは、対応する手法(フラグの設定、量子化ステップ変更子のデータのパックなど)を実行して、ビットストリーム構文に従って量子化ステップ変更子をフォーマットする。代替案では、デコーダおよびエンコーダが、別の構文、たとえば、異なるフラグまたは論理を処理して量子化ステップ変更子をエンコードする構文を使用する。

20

【0262】

図32に、タイルのチャンネルごとの量子化ステップ変更子の検索を示す。その代わりに、タイルを使用しない実施形態で、デコーダが、フレームまたはオーディオデータの他の単位に関してチャンネルごとのステップ変更子を検索する。

【0263】

まず、デコーダは、タイルのチャンネル数が1を超えるかどうかを検査する(3210)。そうでない場合には、オーディオデータがモノラルである。デコーダは、モノラルチャンネルの量子化ステップ変更子に0をセットし(3212)、終了する。

【0264】

マルチチャンネルオーディオについて、デコーダは、複数の変数を初期化する。デコーダは、タイルの量子化ステップ変更子ごとのビット数を示すビット(#BitsPerQ)を得る(3220)。一実施形態では、デコーダが、3ビットを得る。デコーダは、チャンネルカウンタiChannelsDoneに0をセットする(3222)。

30

【0265】

デコーダは、チャンネルカウンタがタイルのチャンネル数より少ないかどうかを検査する(3230)。そうでない場合には、タイルのすべてのチャンネル量子化ステップ変更子が検索されており、デコーダは終了する

その一方で、チャンネルカウンタが、タイルのチャンネル数より少ない場合には、デコーダは、1ビットを入手し(3232)、そのビットを検査して(3240)、現在のチャンネルの量子化ステップ変更子が0であるかどうかを判定する。そうである場合には、デコーダは、現在のチャンネルの量子化ステップ変更子に0をセットする(3242)。

40

【0266】

現在のチャンネルの量子化ステップ変更子が0でない場合には、デコーダは、#BitsPerQが0より大きいかどうかを検査して(3250)、現在のチャンネルの量子化ステップ変更子が1であるかどうかを判定する。そうである場合には、デコーダは、現在のチャンネルの量子化ステップ変更子に1をセットする(3252)。

【0267】

#BitsPerQが0より大きい場合には、デコーダは、ビットストリームの次の#BitsPerQビットを入手し、1を加算し(0の値がより以前の終了条件をトリガす

50

るので)、現在のチャンネルの量子化ステップ変更子にその結果をセットする(3260)。

【0268】

デコーダは、現在のチャンネルの量子化ステップ変更子をセットした後に、チャンネルカウンタを増分し(3270)、チャンネルカウンタがタイルのチャンネル数より少ないかどうかを検査する(3230)。

【0269】

C. 量子化行列のエンコーディングおよびデコーディング

いくつかの実施形態で、エンコーダは、タイルの各チャンネルの量子化行列を計算する。エンコーダは、複数の形で、図1のエンコーダ(100)で使用されるものなどの以前の量子化手法より改善される。量子化行列のロッシイ圧縮に関して、エンコーダは、量子化行列要素の柔軟なステップサイズを使用し、これによって、エンコーダが、量子化行列の要素の分解能を変更できるようになる。この特徴とは別に、エンコーダは、量子化行列の圧縮中に量子化行列値の時間的相関を活用する。

【0270】

前に述べたように、量子化行列は、タイルのチャンネルごとに、パーク周波数帯域(または他の区分された量子化帯域)ごとに1ステップ値の、ステップサイズ配列として働く。エンコーダは、量子化行列を使用して、元の信号に匹敵するスペクトル形状を有するように、再構成されるオーディオ信号を「カラーリング」する。エンコーダは、通常は、音響心理学に基づいて量子化行列を判定し、量子化行列を圧縮して、ビットレートを下げる。量子化行列の圧縮は、ロッシイとすることができる。

【0271】

このセクションに記載の手法は、タイルのチャンネルに関する量子化行列に関して説明される。表記について、 $Q_{m, iChannel, iBand}$ が、帯域*iBand*のチャンネル*iChannel*の量子化行列要素を表すものとする。タイル構成を使用しない実施形態では、エンコーダが、量子化行列要素の柔軟なステップサイズを使用し、かつ/または、圧縮中の量子化行列値の時間的相関を活用することができる。

【0272】

1. マスク情報の柔軟な量子化ステップサイズ

図33に、量子化行列要素の量子化ステップサイズを適応的にセットする一般化された手法(3300)を示す。これによって、エンコーダが、マスク情報を粗くまたは微細に量子化できるようになる。一実施形態では、エンコーダが、タイルのチャンネルごとに(すなわち、タイルの各チャンネルが行列を有する時には行列ごとに)量子化行列要素の量子化ステップサイズをセットする。代替案では、エンコーダが、オーディオシーケンス全体または他のレベルで、タイルごとにまたはフレームごとに、マスク要素の量子化ステップサイズをセットする。

【0273】

エンコーダは、1つまたは複数のマスクの量子化ステップサイズをセットする(3310)ことによって開始する(影響されるマスクの数は、エンコーダが柔軟な量子化ステップサイズを割り当てるレベルに依存する)。一実施形態では、エンコーダが、ある時間の期間にわたって再構成されるオーディオの品質を評価し、その結果に応じて、マスク情報の量子化ステップサイズを1dB、2dB、3dB、または4dBになるように選択する。エンコーダによって評価される品質測定値は、1つまたは複数の前にエンコードされたフレームのNERである。たとえば、全体的な品質が低い場合に、エンコーダは、マスク情報の量子化ステップサイズにより高い値をセットする(3310)ことができる。というのは、量子化行列の分解能が、ビットレートの効率的な使用になっていないからである。その一方で、全体的な品質がよい場合に、エンコーダは、マスク情報の量子化ステップサイズにより低い値をセットする(3310)ことができる。というのは、量子化行列のよりよい分解能によって、知覚される品質が効率的に改善される可能性があるからである。代替案では、エンコーダが、量子化ステップサイズのオープンループ評価で、別の品質

10

20

30

40

50

測定値、異なる期間にわたる評価、および/または他の判断基準を使用する。エンコーダは、マスク情報に異なるまたは追加の量子化ステップサイズを使用することもできる。あるいは、エンコーダが、オープンループ評価をスキップし、その代わりに、ステップサイズの最終的な値に集束するのに結果のクロズドループ評価に頼る。

【0274】

エンコーダは、マスク要素の量子化ステップサイズを使用して1つまたは複数の量子化行列を量子化し(3320)、マルチチャネルオーディオデータに重みをつけ、量子化する。

【0275】

後続の再構成の後に、エンコーダは、NERまたは他の品質測定値を使用して、再構成されたオーディオの品質を評価する(3330)。エンコーダは、再構成されたオーディオの品質が、マスク情報に関する現在の量子化ステップサイズの設定を正当化するものであるかどうかを検査する(3340)。そうでない場合には、エンコーダは、マスク情報の量子化ステップサイズにより高いかより低い値をセットする(3310)ことができる。それ以外の場合には、エンコーダは終了する。代替案では、マスク情報の量子化ステップサイズの1パスオープンループ設定について、エンコーダが、評価(3330)および検査(3340)をスキップする。

10

【0276】

選択の後に、エンコーダは、ビットストリームの適当なレベルでマスク情報の量子化ステップサイズを示す。

20

【0277】

図34に、量子化行列要素の適応量子化ステップサイズを検索する一般化された手法(3400)を示す。したがって、デコーダは、オーディオシーケンス全体または他のレベルについて、タイルのチャンネルごと、タイルごと、またはフレームごとにマスク要素の量子化ステップサイズを変更することができる。

【0278】

デコーダは、1つまたは複数のマスクの量子化ステップサイズを入手する(3410)ことによって開始する(影響されるマスクの数は、エンコーダが柔軟な量子化ステップサイズを割り当てるレベルに依存する)。一実施形態では、量子化ステップサイズが、マスク情報の1dB、2dB、3dB、または4dBである。代替案では、エンコーダおよびデコーダが、マスク情報の異なるまたは追加の量子化ステップサイズを使用する。

30

【0279】

次に、デコーダは、マスク情報の量子化ステップサイズを使用して1つまたは複数の量子化行列を逆量子化し(3420)、マルチチャネルオーディオデータを再構成する。

【0280】

2. 量子化行列の時間予測

図35に、時間予測を使用して量子化行列を圧縮する一般化された手法(3500)を示す。手法(3500)では、エンコーダが、マスク値の時間相関を活用する。これによって量子化行列に関連するビットレートが下がる。

【0281】

図35および36に、オーディオデータのフレームのチャンネルでの量子化行列の時間予測を示す。代替案では、エンコーダが、複数のフレームの間、オーディオの他のシーケンスで、または量子化行列の異なる構成で、時間予測を使用して量子化行列を圧縮する。

40

【0282】

図35を参照すると、エンコーダは、フレームの量子化行列を入手する(3510)。チャンネルの量子化行列は、ウィンドウからウィンドウへと同一のままである傾向があり、予測コーディングのよい候補になる。

【0283】

エンコーダは、時間予測を使用して量子化行列をエンコードする(3520)。たとえば、エンコーダは、図36に示された手法(3600)を使用する。代替案では、エンコ

50

ーダは、時間予測を用いる別の手法を使用する。

【0284】

エンコーダは、圧縮する行列がまだあるかどうかを判定し(3530)、そうでない場合には終了する。それ以外の場合には、エンコーダは、次の量子化行列を入手する。たとえば、エンコーダは、次のフレームの行列がエンコードに使用可能であるかどうかを検査する。

【0285】

図36に、一実施形態で時間予測を使用してチャンネルの量子化行列を圧縮するより詳細な手法(3600)を示す。時間圧縮では、異なるウィンドウサイズのタイルにまたがる再サンプリング処理を使用し、予測残差に対するランレベルコーディングを使用して、ビットレートを下げる。

【0286】

エンコーダは、次に圧縮される量子化行列の圧縮を開始し(3610)、アンカ行列が使用可能であるかどうかを検査する(3620)が、これは、通常は、行列がそのチャンネルの最初の行列であるかどうかに依存する。アンカ行列が使用可能でない場合には、エンコーダは、量子化行列を直接に圧縮する(3630)。たとえば、エンコーダは、量子化行列の要素を差分エンコードし(要素の差分は前の帯域の要素に対するものである)、ハフマンコードを差分に割り当てる。行列の最初の要素(すなわち、帯域0のマスク要素)について、エンコーダは、マスク要素の量子化ステップサイズに依存する予測定数を使用する。

$PredConst = 45 / MaskQuantMultiplier_{iChannel}$
(19)

代替案では、エンコーダが、アンカ行列の別の圧縮手法を使用する。

【0287】

エンコーダは、フレームのチャンネルのアンカ行列として量子化行列をセットする(3640)。エンコーダがタイルを使用する時には、チャンネルのアンカ行列を含むタイルを、アンカタイルと呼ぶことができる。エンコーダは、アンカ行列サイズまたはアンカタイルのタイルサイズを記録するが、これは、異なるサイズを有する行列の予測を形成するのに使用することができる。

【0288】

その一方で、アンカ行列が使用可能である場合には、エンコーダは、時間予測を使用して量子化行列を圧縮する。エンコーダは、チャンネルのアンカ行列に基づいて、量子化行列の予測を計算する(3650)。圧縮される量子化行列が、アンカ行列と同一の数の帯域を有する場合には、予測は、アンカ行列の要素である。しかし、圧縮される量子化行列が、アンカ行列と異なる数の帯域を有する場合には、エンコーダは、アンカ行列を再サンプリングして、予測を計算する。

【0289】

再サンプリング処理では、圧縮される量子化行列のサイズ/現在のタイルサイズと、アンカ行列のサイズ/アンカタイルサイズを使用する。

$MaskPrediction[iBand] = AnchorMask[iScaledBand]$
(20)

ここで、 $iScaledBand$ は、 $iBand$ の代表的な(たとえば平均)周波数を含むアンカ行列帯域である。 $iBand$ は、現在の量子化行列/現在のタイルサイズの項であり、 $iScaledBand$ は、アンカ行列/アンカタイルサイズの項である。

【0290】

図37に、エンコーダがタイルを使用する時の、アンカ行列の再サンプリングの1つの手法を示す。図37には、予測を形成するための、アンカタイルの帯域への現在のタイルの帯域の例の写像(3700)が示されている。現在のタイルの量子化行列の帯域境界の中央の周波数(3720)が、アンカタイルのアンカ行列の周波数に写像(3730)される。マスク予測の値は、写像された周波数が、アンカタイルのアンカ行列の帯域境界(

10

20

30

40

50

3710) に関してどこにあるかに依存してセットされる。代替案では、エンコーダが、チャンネル内の前の量子化行列または他の前の行列に関する時間予測を使用するか、別の再サンプリング手法を使用する。

【0291】

図36に戻って、エンコーダは、予測に関する量子化行列の残差を計算する(3660)。理想的には、予測が、完全であり、残差が、エネルギーを有しない。しかし、必要な場合に、エンコーダは、残差をエンコードする(3670)。たとえば、エンコーダは、予測残差について、ランレベルコーディングまたは別の圧縮手法を使用する。

【0292】

次に、エンコーダは、圧縮される行列がまだあるかどうかを判定し(3680)、そうでない場合に、終了する。それ以外の場合に、エンコーダは、次の量子化行列を入手し(3610)、継続する。

【0293】

図38に、特定のビットストリーム構文による時間予測を使用して圧縮された量子化行列の検索およびデコーディングの手法(3800)を示す。量子化行列は、フレームの単一のタイルのチャンネルに関するものである。図38に、ビットストリームの情報を解析するためにデコーダによって実行される手法(3800)を示し、エンコーダは、対応する手法を実行する。代替案では、デコーダおよびエンコーダが、図38に示されたオプションの1つまたは複数について別の構文、たとえば、異なるフラグまたは異なる順序付けを使用する構文、またはタイルを使用しない構文を使用する。

【0294】

デコーダは、エンコーダがフレームの初めに達したかどうかを検査する(3810)。そうである場合には、デコーダは、そのフレームのすべてのアンカ行列に、セットされていないものとしてマークをつける(3812)。

【0295】

次に、デコーダは、アンカ行列が、次にエンコードされる量子化行列のチャンネルで使用可能であるかどうかを検査する(3820)。アンカ行列が使用可能でない場合には、デコーダは、チャンネルの量子化行列の量子化ステップサイズを入手する(3830)。一実施形態では、デコーダが、1 dB、2 dB、3 dB、または4 dBの値を入手する。

$MaskQuantMultiplier_{iChannel} = getBits(2) + 1$ (21)

【0296】

デコーダは、チャンネルのアンカ行列をデコードする(3832)。たとえば、デコーダは、アンカ行列の差分コーディングされた要素をハフマンデコードし(要素の差分が、前の帯域の要素に対するものである)、要素を再構成する。最初の要素について、デコーダは、エンコーダで使用された予測定数を使用する。

$PredConst = 45 / MaskQuantMultiplier_{iChannel}$ (22)

代替案では、デコーダが、フレームのチャンネルのアンカ行列に別の圧縮解除手法を使用する。

【0297】

デコーダは、フレームのチャンネルのアンカ行列として量子化行列をセットし(3834)、チャンネルの量子化行列の値に、アンカ行列の値をセットする。

$Q_{m, iChannel, iBand} = AnchorMask[iBand]$ (23)

【0298】

デコーダは、アンカタイルのタイルサイズも記録するが、これは、アンカタイルと異なるサイズを有するタイルの行列の予測を形成するのに使用することができる。

【0299】

その一方で、アンカ行列がチャンネルについて使用可能である場合には、デコーダは、時

10

20

30

40

50

間予測を使用して量子化行列を圧縮解除する。デコーダは、チャンネルのアンカ行列に基づいて、量子化行列の予測を計算する(3840)。現在のタイルの量子化行列が、アンカ行列と同一の数の帯域を有する場合には、予測は、アンカ行列の要素である。しかし、現在のタイルの量子化行列が、アンカ行列と異なる数の帯域を有する場合には、エンコーダは、アンカ行列を再サンプリングして、たとえば図37に示された現在のタイルサイズおよびアンカタイルサイズを使用して、予測を入手する。

$$\text{MaskPrediction}[i\text{Band}] = \text{AnchorMask}[i\text{ScaledBand}] \quad (24)$$

【0300】

代替案では、デコーダが、そのチャンネルの前の量子化行列または他の前の行列に対する相対的な時間予測を使用するか、別の再サンプリング手法を使用する。

【0301】

デコーダは、ビットストリームの次のビットを入手し(3842)、ビットストリームに量子化行列の残差が含まれるかどうかを検査する(3850)。現在のタイルのこのチャンネルに関するマスク更新がない場合には、マスク予測残差が0であり、したがって、

$$Q_{m,i\text{Channel},i\text{Band}} = \text{MaskPrediction}[i\text{Band}] \quad (25)$$

になる。

【0302】

その一方で、予測残差がある場合には、デコーダは、たとえばランレベルデコーディングまたは他の圧縮解除手法を使用して、残差をデコードする(3852)。次に、デコーダは、予測に予測残差を加算して(3854)、量子化行列を再構成する。たとえば、加算は、現在のチャンネル*iChannel*の帯域*iBand*に関する要素を入手するために、帯域ごとの単純なスカラ加算である。

【0303】

$$Q_{m,i\text{Channel},i\text{Band}} = \text{MaskPrediction}[i\text{Band}] + \text{MaskPredResidual}[i\text{Band}] \quad (26)$$

その後、デコーダは、現在のタイルのすべてのチャンネルの量子化行列がデコードされたかどうかを検査し(3860)、そうである場合には終了する。そうでない場合には、デコーダは、現在のタイルの次の量子化行列のデコードを継続する。

【0304】

D. 組み合わされた逆量子化および逆重みづけ

デコーダは、必要な量子化および重みづけの情報のすべてを検索したならば、オーディオデータを逆量子化し、逆重みづけする。一実施形態では、デコーダが、逆量子化および逆重みづけを1ステップで実行するが、これを、印刷を明瞭にするために下の2つの式に示す。

【0305】

【数15】

$$\text{CombinedQ} = Q_t + Q_{c,i\text{Channel}} - (\text{Max}(Q_{m,i\text{Channel},*}) - Q_{m,i\text{Channel},i\text{Band}}) \cdot \text{MaskQuantMultiplier}_{i\text{Channel}} \quad (27a)$$

$$y_{iqw}[n] = 10^{\text{CombinedQ}/20} \cdot x_{iqw}[n] \quad (27b)$$

【0306】

ここで、 x_{iqw} は、チャンネル*iChannel*の入力(たとえば、逆マルチチャンネル変換された係数)であり、*n* は、帯域*iBand*の係数インデックスである。 $\text{Max}(Q_{m,i\text{Channel},*})$ は、すべての帯域にわたるチャンネル*iChannel*の最大マスク値である(マスクの最大重みづけ係数と最小重みづけ係数の間の差は、通常は、マスク要素の潜在的な値の範囲よりはるかに小さく、したがって、重みづけ係数ごとの量子化調整の量は、最大値に対して相対的に計算される)。MaskQuantMultipl

10

20

30

40

50

$i e r i c h a n n e l$ は、チャンネル $i C h a n n e l$ の量子化行列のマスク量子化ステップ乗数であり、 $y_{i q w}$ は、このステップの出力である。

【0307】

代替案では、デコーダが、逆量子化および重みづけを、別々にまたは異なる手法を使用して、実行する。

【0308】

V I I . マルチチャンネル後処理

いくつかの実施形態で、図7のデコーダ(700)などのデコーダが、時間領域の再構成されるオーディオサンプルに対するマルチチャンネル後処理を実行する。

【0309】

マルチチャンネル後処理は、多数の異なる目的に使用することができる。たとえば、デコードされるチャンネルの数が、出力のチャンネル数より少ない場合がある(たとえば、コーディングの複雑さまたはバッファ満杯度を下げするために、エンコーダが1つまたは複数の入力チャンネルまたはマルチチャンネル変換されたチャンネルを捨てたので)。その場合に、マルチチャンネル後処理変換を使用して、デコードされたチャンネルの実際のデータに基づいて、1つまたは複数のファントムチャンネルを作成することができる。あるいは、デコードされるチャンネルの数が出力チャンネルの数と等しい場合であっても、提示の任意の空間回転、スピーカ位置の間での出力チャンネルの再写像、または他の立体感あるいは特殊効果に後処理変換を使用することができる。あるいは、デコードされるチャンネルの数が出力チャンネルの数より多い(たとえば、サラウンドサウンドオーディオをステレオ機器で再生する時)場合に、後処理変換を使用して、チャンネルを「折り畳む」ことができる。いくつかの実施形態で、折り曲げられた係数が、潜在的に経時的に変化し、マルチチャンネル後処理が、ビットストリームによって制御される。これらのシナリオおよび応用例の変換行列を、エンコーダによって提供またはシグナリングすることができる。

【0310】

図39に、マルチチャンネル後処理の一般化された手法(3900)を示す。デコーダは、図7に示された手法または他の圧縮解除手法を使用して、エンコードされたマルチチャンネルオーディオデータ(3905)をデコードし(3910)、再構成された時間領域マルチチャンネルオーディオデータ(3915)を作る。

【0311】

デコーダは、次に、時間領域マルチチャンネルオーディオデータ(3915)に対してマルチチャンネル後処理を実行する(3920)。たとえば、エンコーダが、M個のデコードされるチャンネルを作り、デコーダが、Nチャンネルを出力する時に、後処理に、一般的なM対N変換が含まれる。デコーダは、再構成されるM個のコーディングされたチャンネルのそれぞれから1つのM個の同一位置(時間的に)のサンプルをとり、欠けているチャンネル(すなわちエンコーダによって捨てられたN-M個のチャンネル)を、0でパディングする。デコーダは、N個のサンプルに行列 $A_{p o s t}$ をかける。

【0312】

$$Y_{p o s t} = A_{p o s t} \cdot X_{p o s t} \tag{28}$$
ここで、 $X_{p o s t}$ および $Y_{p o s t}$ は、マルチチャンネル後処理へのNチャンネルの入力および出力であり、 $A_{p o s t}$ は、一般的な $N \times N$ 変換行列であり、 $X_{p o s t}$ は、出力ベクトル長Nに一致するように0でパディングされる。

【0313】

行列 $A_{p o s t}$ は、事前に決定された要素を有する行列とすることができ、あるいは、エンコーダによって指定される要素を有する一般的な行列とすることができ、エンコーダは、事前に決定された行列を使用するようにデコーダに知らせる(たとえば、1つまたは複数のフラグビットを用いて)か、一般的な行列の要素をデコーダに送ることができ、あるいは、同一の行列 $A_{p o s t}$ を必ず使用するようにデコーダを構成することができる。行列 $A_{p o s t}$ は、対象または可逆など、特殊な特性を有する必要はない。追加の柔軟性のために、マルチチャンネル後処理を、フレームごとまたは他の基礎でオン/オフにする

10

20

30

40

50

ことができる（この場合に、デコーダは、単位行列を使用して、チャンネルを未変更のままにすることができる）。

【0314】

図40に、図4に示された順序のチャンネルを有する5.1チャンネル再生環境で左チャンネルおよび右チャンネルからファントム中央チャンネルを作成するのに使用される例の行列 $A_{P-center}(4000)$ を示す。例の行列 $A_{P-center}(4000)$ は、他のチャンネルを変更せずに渡す。デコーダは、左、右、サブウーファ、左後ろ、および右後ろのチャンネルから時間的に同位置のサンプルを入手し、中央チャンネルを0でパディングする。その後、デコーダは、6つの入力サンプルに行列 $A_{P-center}(4000)$ をかける。

10

【0315】

【数16】

$$\begin{bmatrix} a \\ b \\ \frac{a+b}{2} \\ d \\ e \\ f \end{bmatrix} = A_{P-Center} \cdot \begin{bmatrix} a \\ b \\ 0 \\ d \\ e \\ f \end{bmatrix} \quad (29)$$

20

【0316】

代替案では、デコーダが、異なる係数を有する行列または異なる個数のチャンネルを使用する。たとえば、デコーダは、行列を使用して、5.1マルチチャンネルオーディオのコーディングされたチャンネルから、7.1チャンネル、9.1チャンネル、または異なる再生環境でのファントムチャンネルを作成する。

【0317】

図41に、フレームごとに変換行列が潜在的に変化するマルチチャンネル後処理の手法(4100)を示す。変換行列の変更は、注意深く扱われない場合に、最終的な出力の可聴雑音（たとえばポンという音）につながる可能性がある。ポンという雑音を導入しないようにするために、デコーダは、ある変換行列から別の変換行列へ、フレームの間に徐々に推移する。

30

【0318】

デコーダは、まず、図7に示された手法または他の圧縮解除手法を使用して、フレームのエンコードされたマルチチャンネルオーディオデータをデコードし(4110)、再構成された時間領域マルチチャンネルオーディオデータを作る。次に、デコーダは、たとえば図42に示されているように、フレームの後処理行列を入手する(4120)。

【0319】

デコーダは、(前のフレームがある場合に)現在のフレームの行列が前のフレームの行列と異なるかどうかを判定する(4130)。現在の行列が同一であるか、前の行列が存在しない場合には、デコーダは、現在のフレームの再構成されたオーディオサンプルに行列を適用する(4140)。そうでない場合には、デコーダは、現在のフレームの再構成されたオーディオサンプルにブレンドされた変換行列を適用する(4150)。ブレンド関数は、実施形態に依存する。一実施形態では、現在のフレームのサンプル i で、デコーダが、短期間ブレンドされた行列 $A_{post,i}$ を使用する。

40

【0320】

【数17】

$$A_{post,i} = \frac{NumSamples - i}{NumSamples} A_{post,prev} + \frac{i}{NumSamples} A_{post,current} \quad (30)$$

【0321】

50

ここで、 $A_{post, prev}$ および $A_{post, current}$ は、前のフレームおよび現在のフレームの後処理行列であり、 $NumSamples$ は、現在のフレームのサンプル数である。代替案では、デコーダが、別のブレンディング関数を使用して、後処理変換行列の不連続性を平滑化する。

【0322】

デコーダは、フレームごとに手法(4100)を繰り返す。代替案では、デコーダは、他の基礎でマルチチャンネル後処理を変更する。

【0323】

図42に、特定のビットストリーム構文によるマルチチャンネル後処理の変換行列を識別し、検索する手法(4200)を示す。この構文を用いると、事前定義された変換行列ならびにマルチチャンネル後処理のカスタム行列の指定が可能になる。図42には、ビットストリームを解析するためにデコーダによって実行される手法(4200)が示され、エンコーダは、対応する手法(フラグの設定、要素のデータのパックなど)を実行して、ビットストリーム構文に従って変換行列をフォーマットする。代替案では、デコーダおよびエンコーダが、図42に示されたオプションの1つまたは複数に、別の構文、たとえば、異なるフラグまたは異なる順序付けを使用する構文を使用する。

10

【0324】

まず、デコーダは、チャンネルの数 $\#Channels$ が1より大きいかどうかを判定する(4210)。 $\#Channels$ が1である場合には、オーディオデータはモノラルであり、デコーダは、単位行列を使用する(4212)(すなわち、マルチチャンネル後処理自体を実行しない)。

20

【0325】

その一方で、 $\#Channels > 1$ の場合には、デコーダは、ビットストリームの次のビットと等しくなるように一時値 $iTmp$ をセットする(4220)。次に、デコーダは、一時値の値を検査するが(4230)、この値によって、デコーダが単位行列を使用しなければならない(4232)か否かが示される。

【0326】

デコーダが、マルチチャンネルオーディオに単位行列以外の何かを使用する場合には、デコーダは、ビットストリームの次のビットと等しくなるように一時値 $iTmp$ をセットする(4240)。次に、デコーダは、一時値の値を検査するが(4250)、この値によって、デコーダが事前定義のマルチチャンネル変換行列を使用(4252)しなければならないか否かが示される。デコーダが、事前定義の行列を使用する(4252)場合には、デコーダは、複数の使用可能な事前定義の行列のどれをデコーダが使用しなければならないかを示す1つまたは複数の追加ビットをビットストリーム(図示せず)から入手することができる。

30

【0327】

デコーダが、事前定義の行列を使用しない場合には、デコーダは、カスタム行列をデコードするために、さまざまな一時値を初期化する。デコーダは、終了した係数のカウンタ $iCoefsDone$ に0をセットし(4260)、行列の要素数($\#Channels^2$)と等しくなるように、デコードする係数の数 $\#CoefsToDo$ をセットする(4262)。特定の特性(たとえば対照)を有することが既知の行列について、デコードされる係数の数を減らすことができる。次に、デコーダは、すべての係数がビットストリームから検索されたかどうかを判定し(4270)、そうである場合には終了する。そうでない場合には、デコーダは、行列の次の要素の値 $A[iCoefsDone]$ を入手し(4272)、 $iCoefsDone$ を増分する(4274)。要素がコーディングされ、ビットストリームにパックされる形は、実装依存である。図42では、構文によって、変換行列の要素ごとに4ビットの精度が可能であり、各要素の絶対値が、1以下である。他の実施形態では、要素ごとの精度が、異なり、エンコーダおよびデコーダが、変換行列の冗長性のパターンを活用する圧縮を使用し、かつ/または構文が、他の形で異なる。

40

【0328】

50

好ましい実施形態に関して本発明の原理を説明し、示したが、説明された実施形態を、そのような原理から逸脱せずに、配置および詳細において修正できることを諒解されたい。本明細書に記載のプログラム、処理、または方法は、特に示されない限り、コンピューティング環境の特定のタイプに関係せず、制限されないことを理解されたい。さまざまなタイプの汎用コンピューティング環境および特殊化されたコンピューティング環境は、本明細書に記載の教示による動作と共に使用されるか、その動作を実行することができる。説明された実施形態の、ソフトウェアで示された要素は、ハードウェアで実施することができ、逆も同様である。

【0329】

本発明の原理を適用できる多数の可能な実施形態に鑑みて、本発明者は、そのような実施形態のすべてを、請求項およびその均等物の範囲および趣旨に含めることができると主張する。

【図面の簡単な説明】

【0330】

【図1】従来技術によるオーディオエンコーダのブロック図である。

【図2】従来技術によるオーディオデコーダのブロック図である。

【図3a】従来技術によるステレオオーディオデータのフレームのウィンドウ構成を示す図である。

【図3b】従来技術によるステレオオーディオデータのフレームのウィンドウ構成を示す図である。

【図3c】従来技術によるステレオオーディオデータのフレームのウィンドウ構成を示す図である。

【図4】5.1チャンネル/スピーカ構成の6つのチャンネルを示す図である。

【図5】説明されている本実施形態を実施することができる適切なコンピューティング環境のブロック図である。

【図6】説明されている本実施形態を実施することができるオーディオエンコーダのブロック図である。

【図7】説明されている本実施形態を実施することができるオーディオデコーダのブロック図である。

【図8】マルチチャンネル前処理の一般化された手法を示す流れ図である。

【図9a】マルチチャンネル前処理の例の行列を示す図である。

【図9b】マルチチャンネル前処理の例の行列を示す図である。

【図9c】マルチチャンネル前処理の例の行列を示す図である。

【図9d】マルチチャンネル前処理の例の行列を示す図である。

【図9e】マルチチャンネル前処理の例の行列を示す図である。

【図10】フレームごとに変換行列が潜在的に変化するマルチチャンネル前処理の手法を示す流れ図である。

【図11a】マルチチャンネルオーディオの例のタイル構成を示す図である。

【図11b】マルチチャンネルオーディオの例のタイル構成を示す図である。

【図12】マルチチャンネルオーディオのタイルを構成する一般化された手法を示す流れ図である。

【図13】特定のビットストリーム構文によるマルチチャンネルオーディオの並列のタイル構成およびタイル情報送出の手法を示す流れ図である。

【図14】知覚的重みづけの後にマルチチャンネル変換を実行する一般化された手法を示す流れ図である。

【図15】逆知覚的重みづけの前に逆マルチチャンネル変換を実行する一般化された手法を示す流れ図である。

【図16】一実施形態でマルチチャンネル変換についてタイル内のチャンネルをグループ化する手法を示す流れ図である。

【図17】特定のビットストリーム構文によるビットストリームからのタイルのチャンネル

10

20

30

40

50

グループ情報およびマルチチャンネル変換情報の検索の手法を示す流れ図である。

【図18】一実施形態でマルチチャンネル変換にチャンネルグループの周波数帯域を選択的に含める手法を示す流れ図である。

【図19】特定のビットストリーム構文によるビットストリームからのタイルのチャンネルグループに関するマルチチャンネル変換の帯域オン/オフ情報を検索する手法を示す流れ図である。

【図20】より単純なマルチチャンネル変換の階層を使用してマルチチャンネル変換をエミュレートする一般化された手法を示す流れ図である。

【図21】マルチチャンネル変換の例の階層を示す図である。

【図22】特定のビットストリーム構文によるビットストリームからのチャンネルグループに関するマルチチャンネル変換の階層の情報を検索する手法を示す流れ図である。

【図23】複数の使用可能なタイプの中からマルチチャンネル変換タイプを選択する一般化された手法を示す流れ図である。

【図24】複数の使用可能なタイプの中からマルチチャンネル変換タイプを検索し、逆マルチチャンネル変換を実行する手法を示す流れ図である。

【図25】特定のビットストリーム構文によるビットストリームからチャンネルグループに関するマルチチャンネル変換情報を検索する手法を示す流れ図である。

【図26】マルチチャンネル変換行列を表すギブンス回転の回転行列の一般形を示す図である。

【図27a】マルチチャンネル変換行列を表すギブンス回転の例の回転行列を示す図である。

【図27b】マルチチャンネル変換行列を表すギブンス回転の例の回転行列を示す図である。

【図27c】マルチチャンネル変換行列を表すギブンス回転の例の回転行列を示す図である。

【図28】量子化されたギブンス因数分解回転を使用してマルチチャンネル変換行列を表す一般化された手法を示す流れ図である。

【図29】特定のビットストリーム構文によるビットストリームからチャンネルグループの汎用ユニタリ変換の情報を検索する手法を示す流れ図である。

【図30】特定のビットストリーム構文によるビットストリームからタイルの全体的なタイル量子化係数を検索する手法を示す流れ図である。

【図31】マルチチャンネルオーディオデータのチャンネルごとの量子化ステップ変更子を計算する一般化された手法を示す流れ図である。

【図32】特定のビットストリーム構文によるビットストリームからチャンネルごとの量子化ステップ変更子を検索する手法を示す流れ図である。

【図33】量子化行列要素の量子化ステップサイズを適応式にセットする一般化された手法を示す流れ図である。

【図34】量子化行列要素の適応量子化ステップサイズを検索する一般化された手法を示す流れ図である。

【図35】時間予測を使用して量子化行列を圧縮する手法を示す流れ図である。

【図36】時間予測を使用して量子化行列を圧縮する手法を示す流れ図である。

【図37】量子化行列要素の予測のための帯域の写像を示す図である。

【図38】特定のビットストリーム構文による時間予測を使用して圧縮された量子化行列の検索およびデコーディングの手法を示す流れ図である。

【図39】マルチチャンネル後処理の一般化された手法を示す流れ図である。

【図40】マルチチャンネル後処理の例の行列を示す図である。

【図41】フレームごとに変換行列が潜在的に変化するマルチチャンネル後処理の手法を示す流れ図である。

【図42】特定のビットストリームによるマルチチャンネル後処理の変換行列を識別し、検索する手法を示す流れ図である。

10

20

30

40

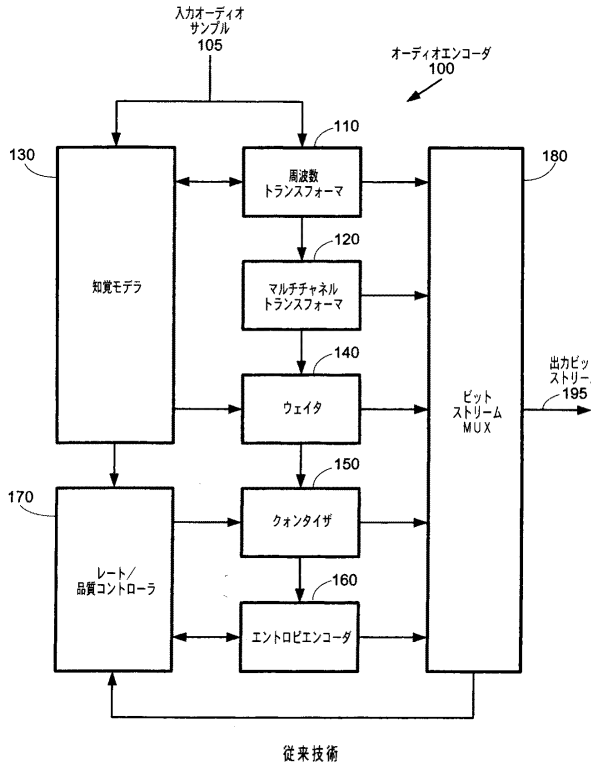
50

【符号の説明】

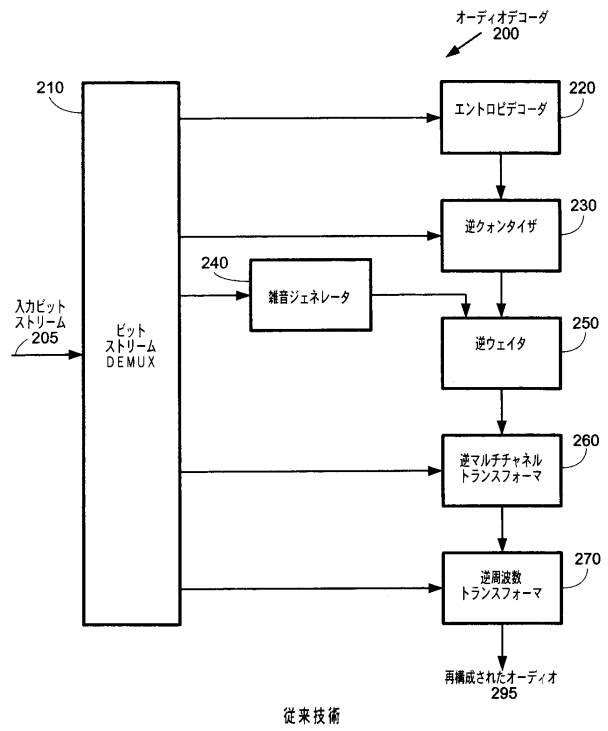
【0331】

400	5.1チャンネル/スピーカ配置行列	
500	コンピューティング環境	
510	処理ユニット	
520	メモリ	
570	通信接続	
550	入力デバイス	
560	出力デバイス	
540	ストレージ	10
600	オーディオエンコーダ	
605	入力オーディオサンプル	
608	セレクタ	
610	マルチチャンネルプリプロセッサ	
620	パーティショナ/タイルコンフィギュアラ	
630	周波数トランスフォーマ	
640	知覚モデラ	
642	量子化帯域ウェイト	
644	チャンネルウェイト	
690	MUX	20
695	出力ビットストリーム	
650	マルチチャンネルトランスフォーマ	
672	ミックスド/ピュアロスレスコーダ	
674	エントロピエンコーダ	
680	レート/品質コントローラ	
660	クオンタイザ	
670	エントロピエンコーダ	
700	オーディオデコーダ	
705	入力ビットストリーム	
710	DEMUX	30
730	タイル構成デコーダ	
720	エントロピデコーダ	
740	逆マルチチャンネルトランスフォーマ	
750	逆クオンタイザ/ウェイト	
760	逆周波数トランスフォーマ	
770	オーバーラップ/アダー	
722	ミックスド/ピュアロスレスデコーダ	
780	マルチチャンネルポストプロセッサ	
795	再構成されたオーディオ	
805	時間領域マルチチャンネルオーディオデータ	40
815	時間領域マルチチャンネル変換されたオーディオデータ	
825	エンコードされたマルチチャンネルオーディオデータ	

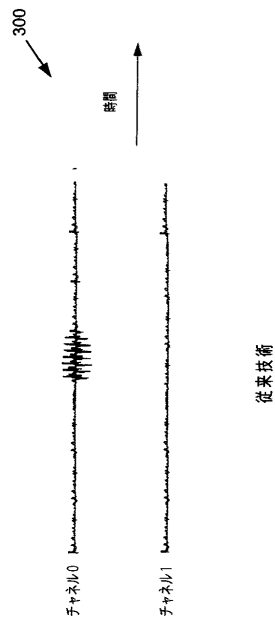
【図1】



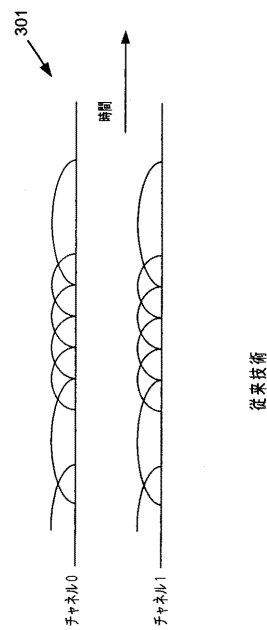
【図2】



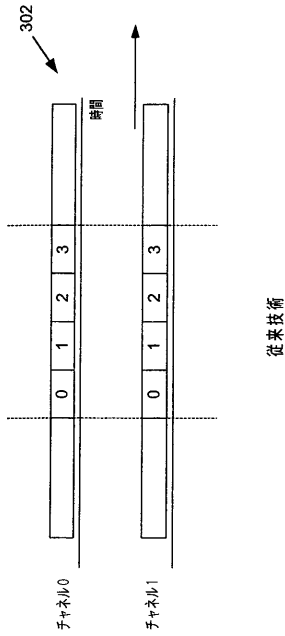
【図3a】



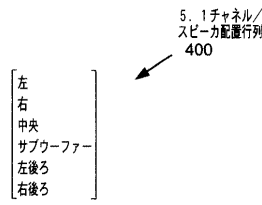
【図3b】



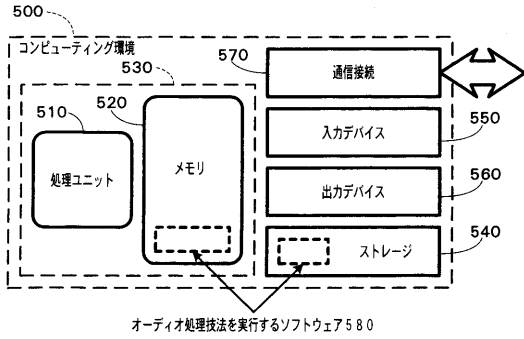
【図3c】



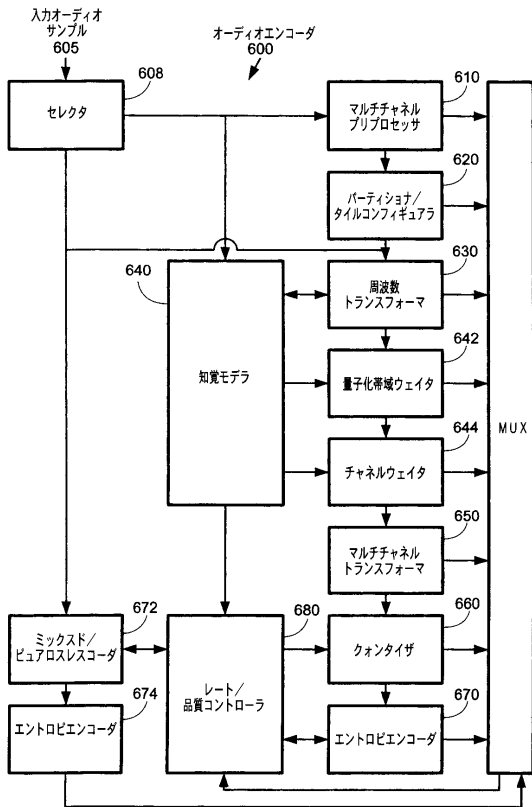
【図4】



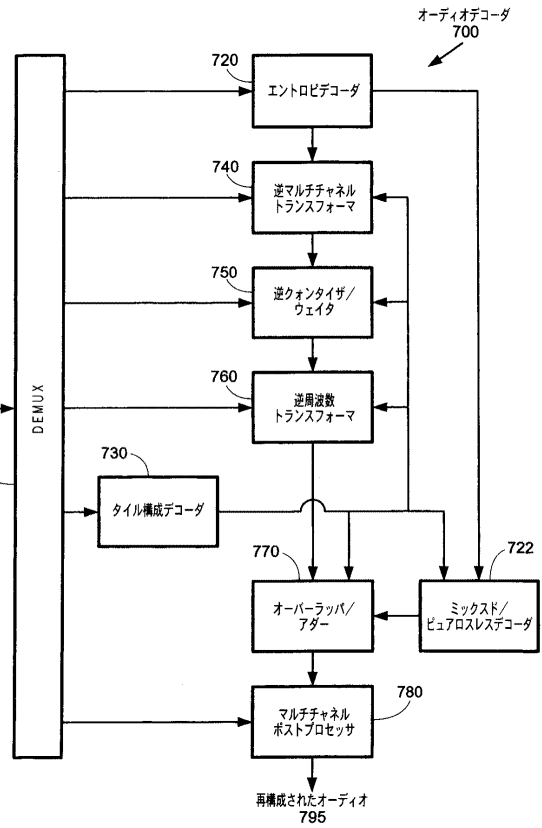
【図5】



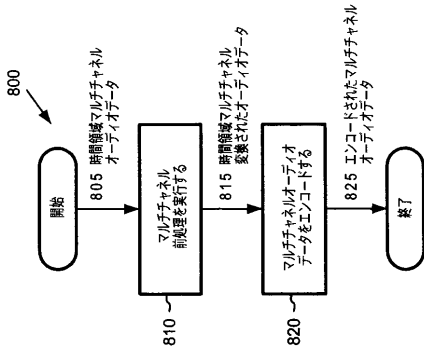
【図6】



【図7】



【図8】



【図9a】

$$A_{low} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad 900$$

【図9b】

$$A_{inter,1} = \begin{bmatrix} \left(\frac{1}{1+0.5\alpha}\right) & 0 & \left(\frac{0.5\alpha}{1+0.5\alpha}\right) & 0 & 0 & 0 \\ 0 & \left(\frac{1}{1+0.5\alpha}\right) & \left(\frac{0.5\alpha}{1+0.5\alpha}\right) & 0 & 0 & 0 \\ \left(\frac{\alpha}{1+2\alpha}\right) & \left(\frac{\alpha}{1+2\alpha}\right) & \left(\frac{1}{1+2\alpha}\right) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \left(\frac{1}{1+\alpha}\right) & 0 \\ 0 & 0 & 0 & 0 & \left(\frac{1}{1+\alpha}\right) & \left(\frac{\alpha}{1+\alpha}\right) \\ 0 & 0 & 0 & 0 & \left(\frac{\alpha}{1+\alpha}\right) & \left(\frac{1}{1+\alpha}\right) \end{bmatrix} \quad 901$$

【図9c】

$$A_{high,1} = \begin{bmatrix} \left(\frac{1}{1.5}\right) & 0 & \left(\frac{0.5}{1.5}\right) & 0 & 0 & 0 \\ 0 & \left(\frac{1}{1.5}\right) & \left(\frac{0.5}{1.5}\right) & 0 & 0 & 0 \\ \left(\frac{1}{3}\right) & \left(\frac{1}{3}\right) & \left(\frac{1}{3}\right) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 \end{bmatrix} \quad 902$$

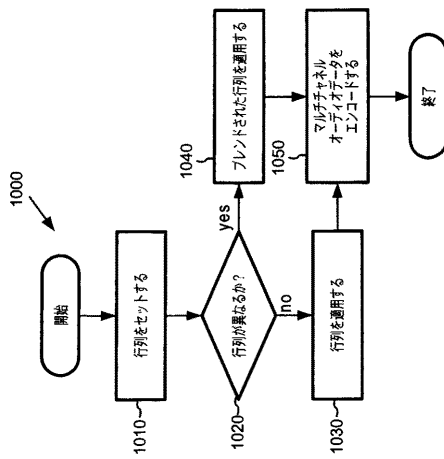
【図9d】

$$A_{inter,2} = \begin{bmatrix} \left(\frac{1}{1+0.5\alpha}\right) & 0 & \left(\frac{0.5\alpha}{1+0.5\alpha}\right) & 0 & 0 & 0 \\ 0 & \left(\frac{1}{1+0.5\alpha}\right) & \left(\frac{0.5\alpha}{1+0.5\alpha}\right) & 0 & 0 & 0 \\ 0.5\alpha & 0.5\alpha & 1-\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \left(\frac{1}{1+\alpha}\right) & \left(\frac{\alpha}{1+\alpha}\right) \\ 0 & 0 & 0 & 0 & \left(\frac{\alpha}{1+\alpha}\right) & \left(\frac{1}{1+\alpha}\right) \end{bmatrix} \quad 903$$

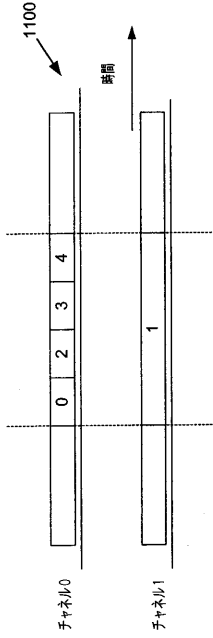
【図9e】

$$A_{high,2} = \begin{bmatrix} \left(\frac{1}{1.5}\right) & 0 & \left(\frac{0.5}{1.5}\right) & 0 & 0 & 0 \\ 0 & \left(\frac{1}{1.5}\right) & \left(\frac{0.5}{1.5}\right) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 \end{bmatrix} \quad 904$$

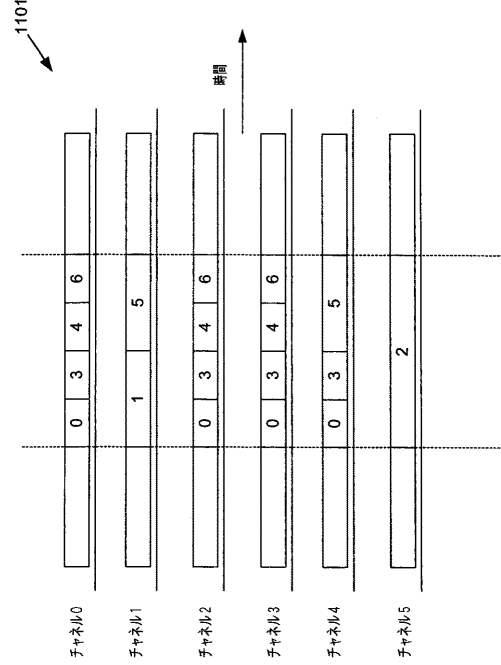
【図10】



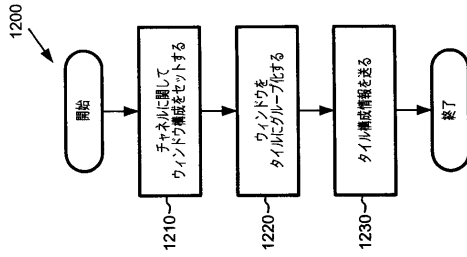
【図 1 1 a】



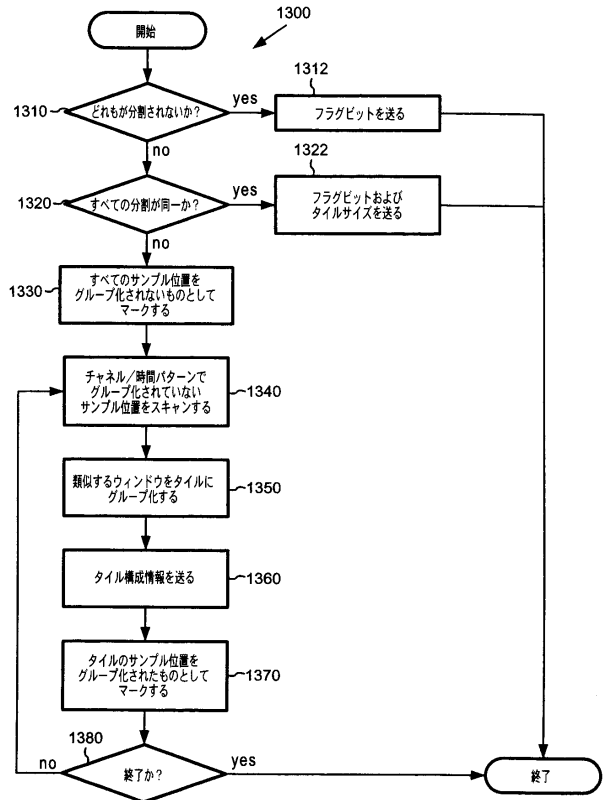
【図 1 1 b】



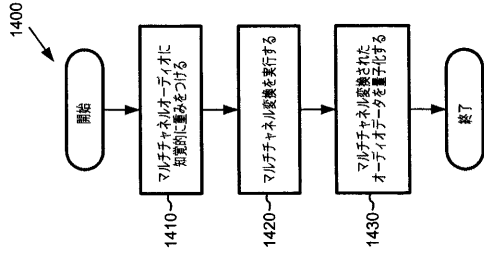
【図 1 2】



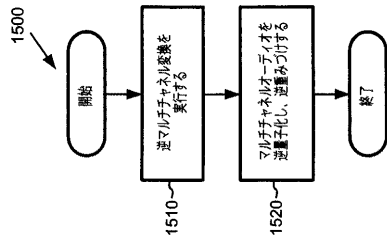
【図 1 3】



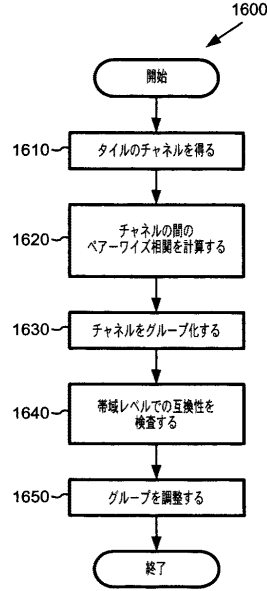
【図 14】



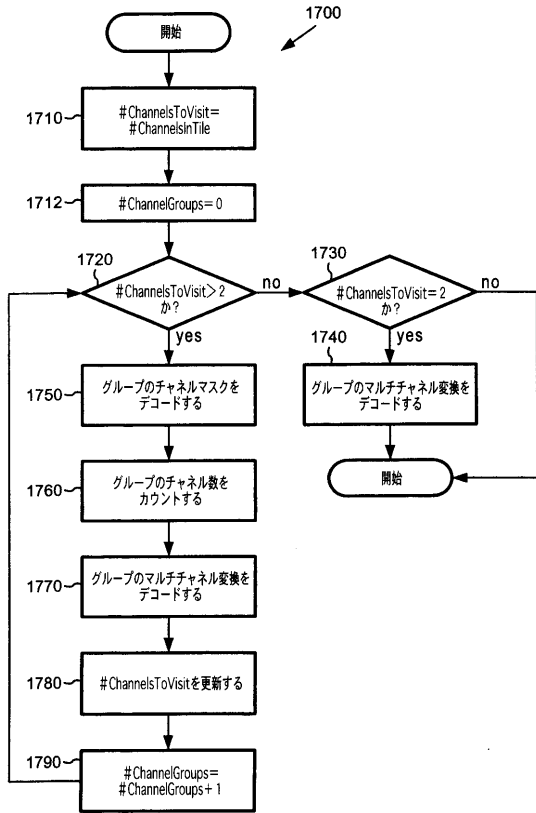
【図 15】



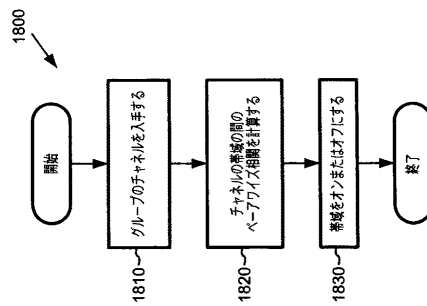
【図 16】



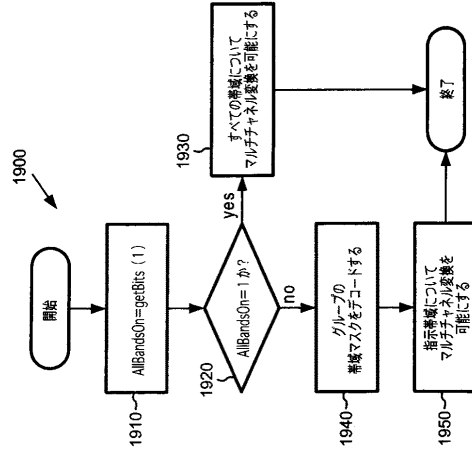
【図 17】



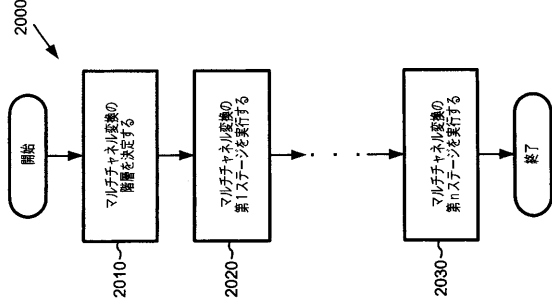
【図 18】



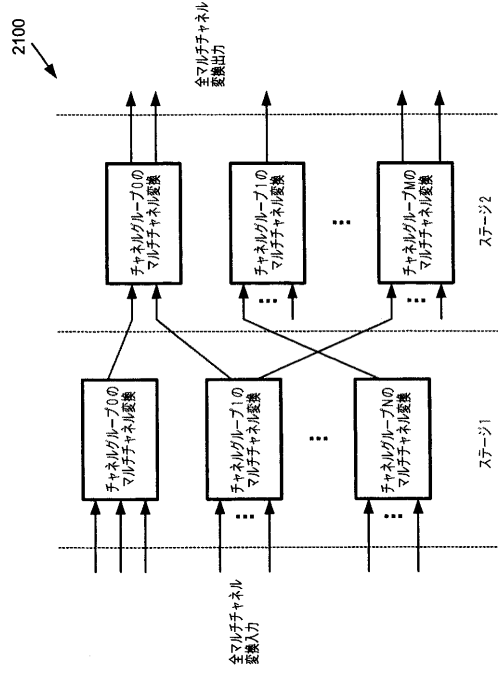
【図 19】



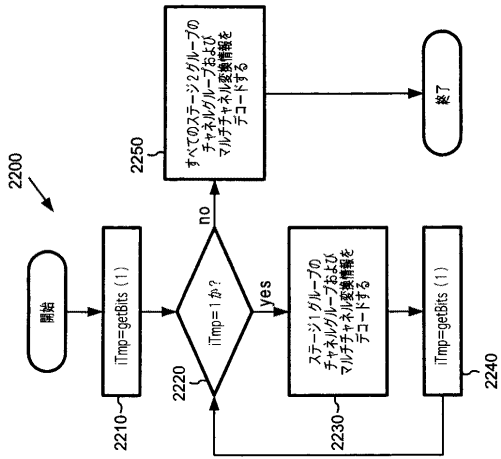
【図 20】



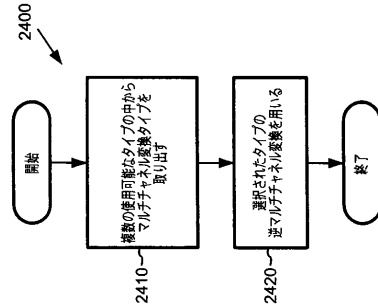
【図 21】



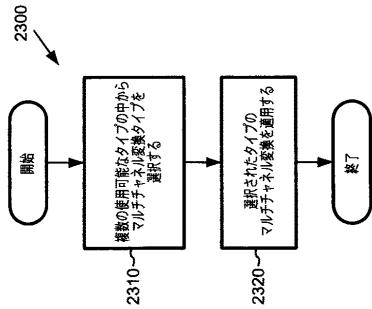
【図 22】



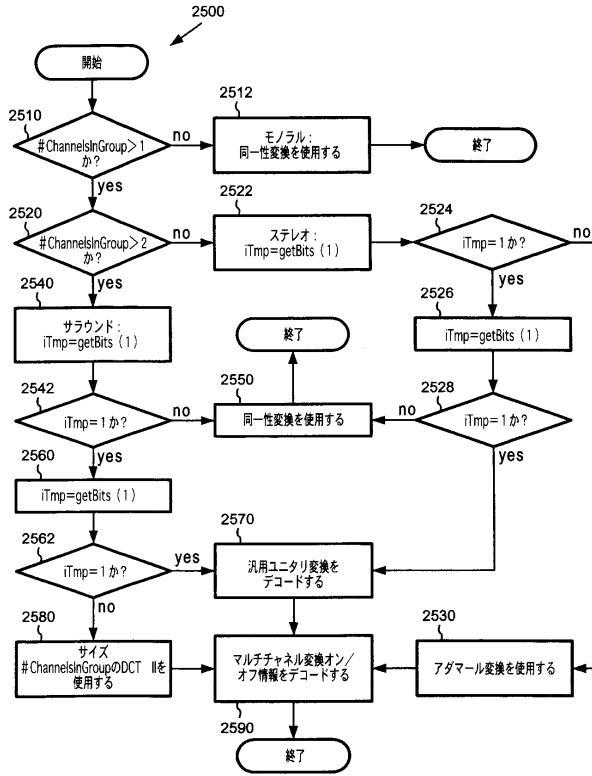
【図 24】



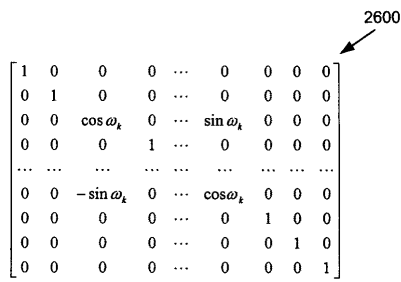
【図 23】



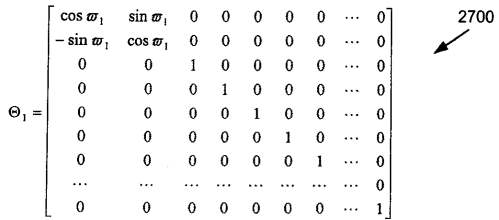
【図25】



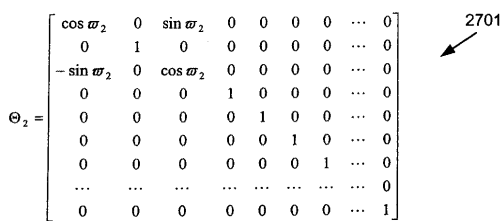
【図26】



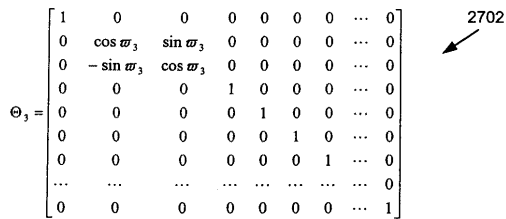
【図27a】



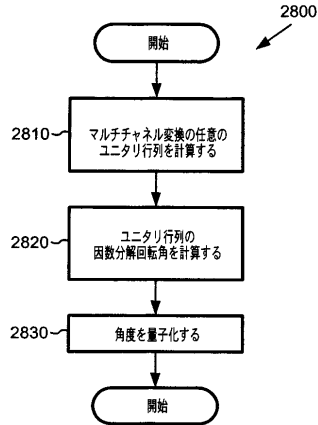
【図27b】



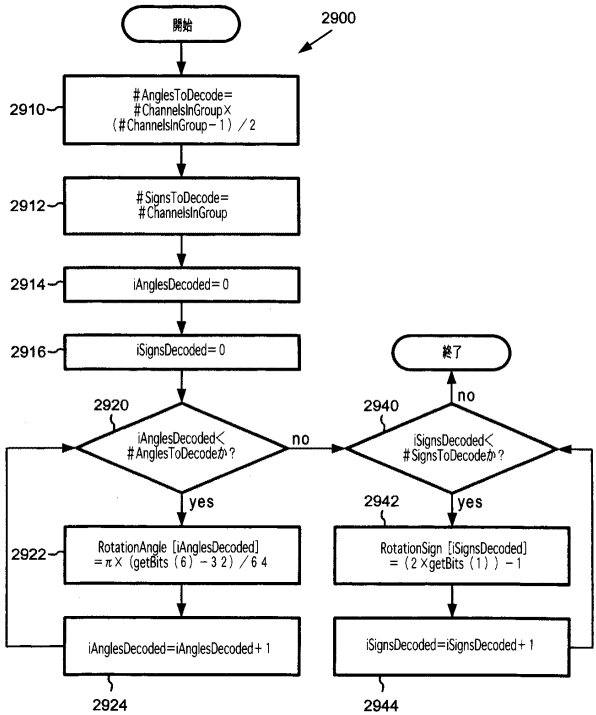
【図27c】



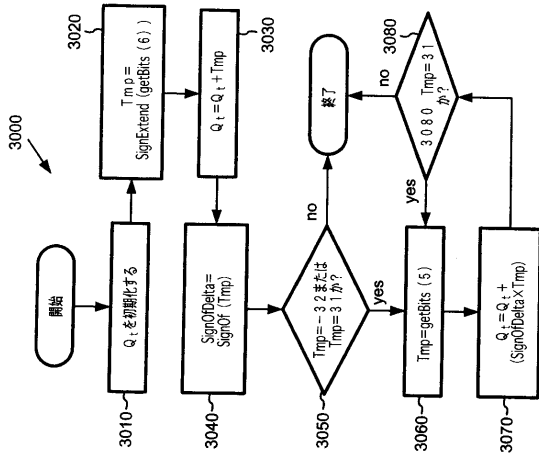
【図28】



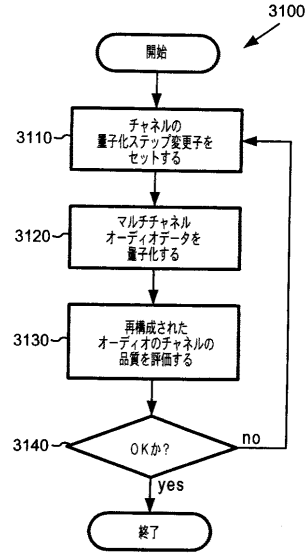
【図29】



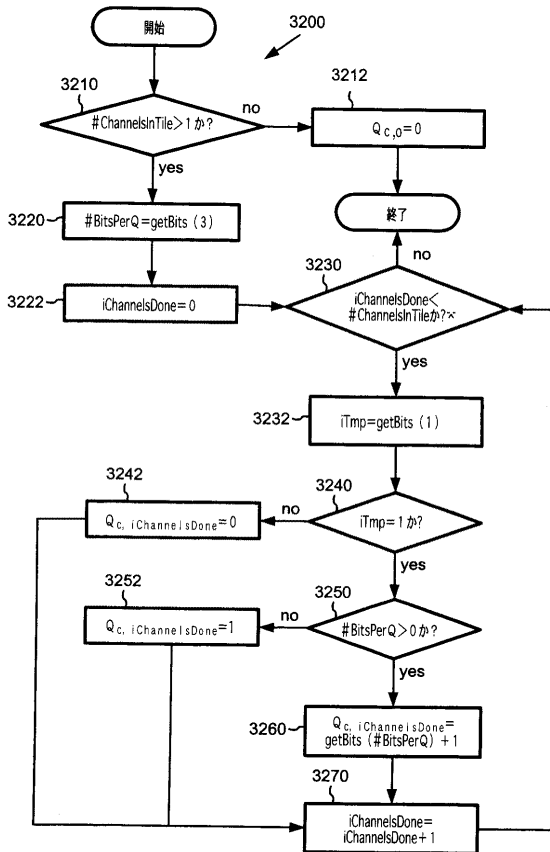
【図30】



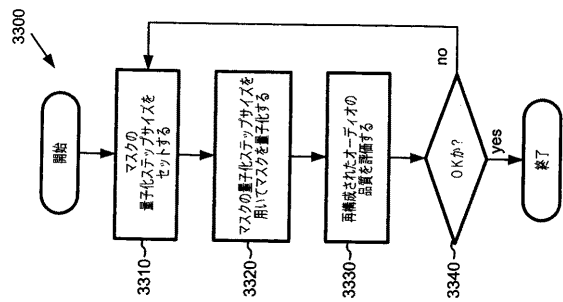
【図31】



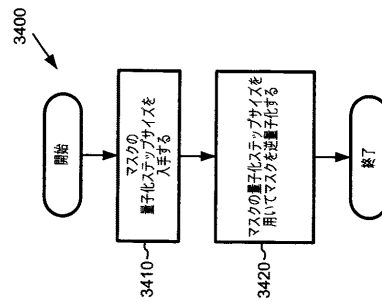
【図32】



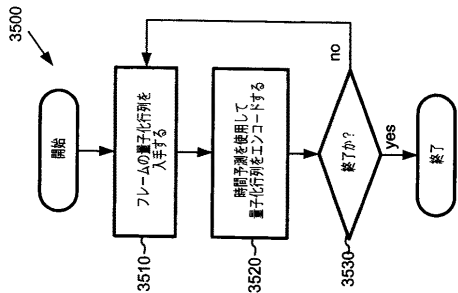
【図33】



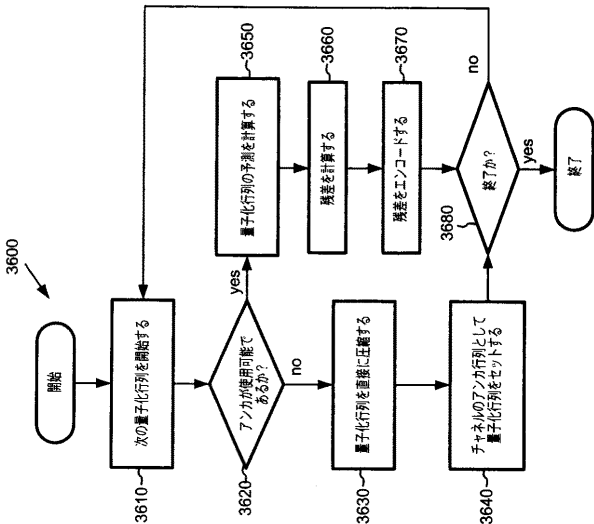
【図34】



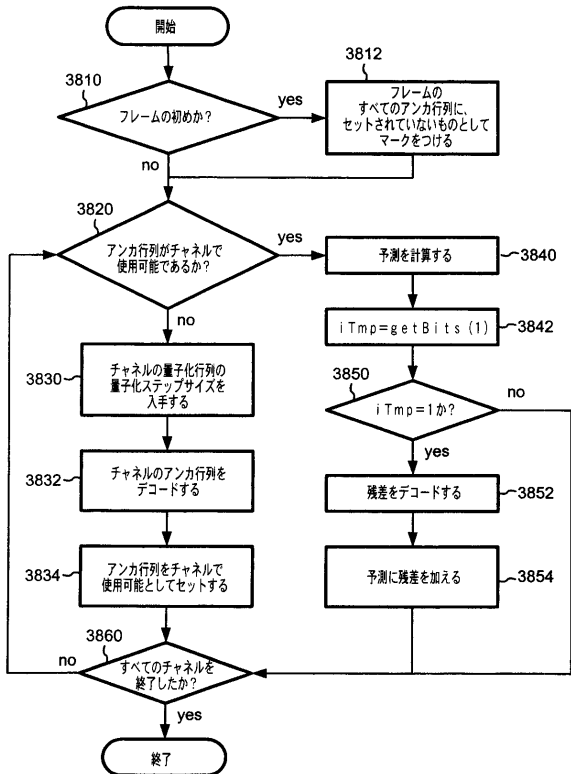
【図 35】



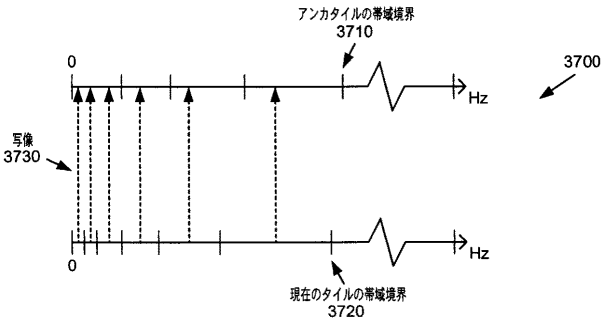
【図 36】



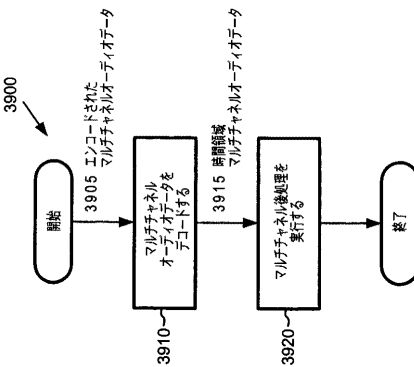
【図 38】



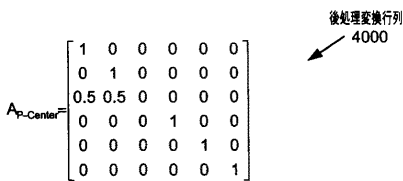
【図 37】



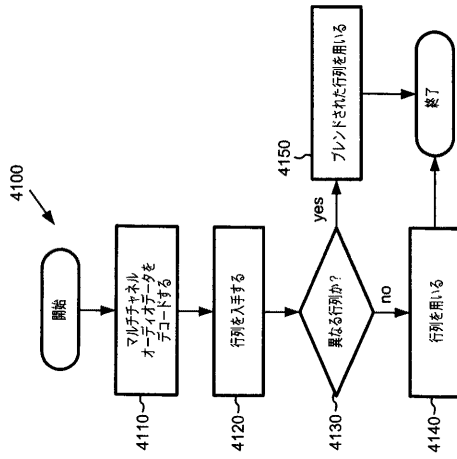
【図 39】



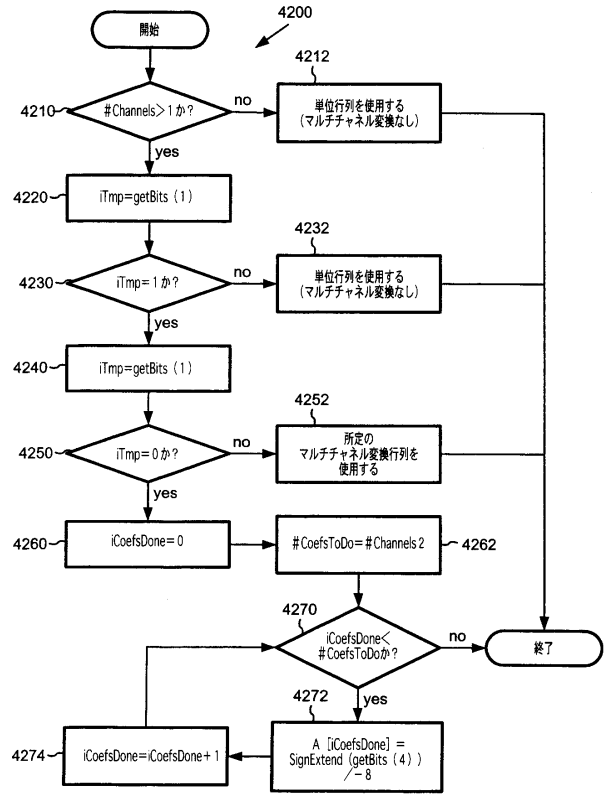
【図 40】



【図 4 1】



【図 4 2】



フロントページの続き

(72)発明者 ウエイ - ゲ チェン

アメリカ合衆国 98029 ワシントン州 イサコア サウスイースト 37 ストリート 2
4635

審査官 山下 剛史

(56)参考文献 特開2001-44844(JP,A)

国際公開第00/60746(WO,A1)

特表平9-500503(JP,A)

特開平6-149292(JP,A)

特開平6-075590(JP,A)

国際公開第95/02925(WO,A1)

(58)調査した分野(Int.Cl., DB名)

G10L 19/00 - 19/14

H04B 14/04

H03M 7/30